ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
A MASTER'S DEGREE IN MECHANICAL ENGINEERING
M.Eng.

BY
Oscar Alejandro URBINA RIVAS

NUMERICAL SIMULATION OF THE FLOW AROUND A WIND TURBINE USING OPENFOAM

MONTREAL, SEPTEMBER 18 2014

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Prof. Christian Masson, Research Director
Mechanical Engineering Department

Prof. Simon Joncas, Committee President
Automated Manufacturing Engineering Department

Prof. Christophe Sibuet Watters, External Examiner
Cégep du Vieux Montréal

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

ON SEPTEMBER 3 2014

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

Firstly, I would like to thank professor Christian Masson for being my research director, for his patience, and for providing the knowledge, encouragement and resources I needed to be able to complete this project. To the Wind Energy Strategic Network (WESNet) for the financial support it offers to numerous graduate students including myself.

Thanks to Jörn Nathan, Mary Bautista, Hugo Olivares, Eric Jeannotte, Alex Flores Madariaga, Yann-Aël Muller, Hajer Ben Younes, David Lapointe-Thériault, Chérif Lateb, and Nicolas Gasset for the interesting conversations and invaluable advice that had a positive impact in the completion of this project. To all my friends for their constant encouragement and friendship throughout the years.

Lastly, I would like to thank my family. My father for being an amazing source of inspiration, for encouraging me to exceed my limits and to strive to be not only a better professional but a better person. To my mother whose love and kindness was a constant source of encouragement in difficult times, to my sister Mercedes, my brother Luis, and my brother Ricardo and his family for their patience, love and for being there for me when I needed it most.

NUMERICAL SIMULATION OF THE FLOW AROUND A WIND TURBINE USING OPENFOAM

Oscar Alejandro URBINA RIVAS

ABSTRACT

The vortical wake created by wind turbines has been the subject of aerodynamics studies for many years, and multiple prediction models have been developed over the years in a search for a simple yet accurate means of predicting blade loading and their effect on flow. In recent years, the development of computational fluid dynamics (CFD) has given rise to new simulation methods for wind turbines. Amongst the more popular methods for modelling wind turbine rotors, we note the use of volume forces, which are able to correctly model the far wake but fail to accurately reproduce near wake characteristics. As an alternative, the Actuator Surface (AS) model developed by Sibuet Watters and Masson (2010) and its implementation within the differential Navier-Stokes equations in the CFD program OpenFOAM is presented in this thesis.

The AS is an infinitely thin surface carrying velocity discontinuities tangent to the surface and an attached system of forces given by a Kutta-Joukowski relation that generate a pressure discontinuity normal to the surface. The AS can also be described as a porous vortex sheet that generates lift forces linked to its circulation characteristics. Using the finite volume method (FVM) in OpenFOAM, the AS has been implemented in two-dimensional and threedimensional applications.

The AS has been validated against cases where an analytical solution is available such as an infinite plane, a light loaded actuator disk and a 2D segment of velocity discontinuity. The AS is then applied to the study of a finite wing in translation, where it has been verified that the AS accurately predicts characteristics such as the induced angle which is compared with Prandtl's lifting line theory. The final application is the study of a wind turbine where blade element analysis has been used to determine AS loading. In the case of the wind turbine, results have been compared with experimental measurements and it has been demonstrated that the AS is an appropriate method for turbine wake predictions, being able to model both near and far wake characteristics.

Keywords: Actuator Surface, wind turbine aerodynamics, vortical wake, CFD, OpenFOAM.

SIMULATION NUMÉRIQUE DES ÉCOULEMENTS AUTOUR D'UNE ÉOLIENNE EN OPENFOAM

Oscar Alejandro URBINA RIVAS

RÉSUMÉ

Depuis plusieurs années le sillage des éoliennes est un sujet d'études en aérodynamique. À travers les années plusieurs modèles de prédiction ont été développés en cherchant un moyen simple et efficace de prédiction des charges des pales d'une éolienne et de ses effets sur l'écoulement. Dans les dernières années, avec le développement de la mécanique des fluides assistée par ordinateur (CFD), des nouvelles méthodes de simulation des éoliennes sont apparues; parmi les méthodes plus populaires de modélisation des rotors d'éolienne,on trouve l'utilisation des forces volumiques, capables de bien modéliser le sillage lointain mais avec une représentation moins fine du sillage proche. Comme alternative, la méthode de la surface actuatrice (SA) développée par Sibuet Watters and Masson (2010) et son implémentation dans les équations différentielles de Navier-Stokes avec le logiciel OpenFOAM sont présentées dans ce mémoire.

La SA est une surface infiniment mince porteuse de discontinuités de vitesses tangentielles à la surface et un système des forces attachées donné par une relation du type Kutta-Jukowski, qui génère une discontinuité de pression dans la direction normale à la surface. La SA peut aussi être décrite comme une nappe tourbillonnaire poreuse qui génère des forces de portance reliées à ses caractéristiques de circulation. Des applications bidimensionnelles et tridimensionnelles ont été implémentés en utilisant une méthode de volumes finis (MVF) dans OpenFOAM.

La SA a été validée avec des cas dont une solution analytique est disponible, comme le cas du plan infini, le disque actuateur légèrement chargé ou un segment bidimensionnel porteur d'une discontinuité de vitesse. La SA est aussi appliquée pour l'étude d'une aile en translation, l'étude de ce cas a démontré que la SA peut réproduire correctement des caractéristiques telles que l'angle induit, qui a été comparé avec la théorie de la ligne portante de Prandtl. L'application finale est l'étude d'une éolienne, l'analyse d'élément de pale ayant été utilisée pour calculer la charge des pales. Les résultats sont comparés avec des données expérimentales et ils montrent que la SA est une méthode appropriée pour la prédiction des sillages, capable de bien modéliser les caractéristiques du sillage proche et lointain.

Mot-clés : Surface actuatrice, aérodynamique des éoliennes, sillage turbillonnaire, CFD, OpenFOAM

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ABBREVIATIONS

AR	Aspect Ratio
AS	Actuator Surface
BEM	Blade Element Momentum
CanWEA	Canadian Wind Energy Association
CFD	Computational Fluid Dynamics
ÉTS	École de Technologie Supérieure
FVM	Finite Volume Method
IEA	International Energy Agency
LDV	Laser Doppler Velocimetry
MEXICO	Model experiments in Controlled Conditions
MRF	Multiple Reference Frame
NREL	National Renewable Energy Laboratory
OpenFOAM	Open Field Operation and Manipulation
PIV	Particle Image Velocimetry
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SRF	Single Reference Frame
TUDelft	Technical University of Delft

LIST OF SYMBOLS

A	Outer area of a control volume (m^2)
B	Domain side length (m)
b	Wing side length (m)
C_P	Power coefficient
C_T	Thrust coefficient
c	Chord (m)
cl	Lift coefficient
D	Diameter (m)
f	Force (N/m^2)
i	Random node
L	Lift (N)
l	Segment length (m)
n	Normal vector
P	Random point
p	Pressure (Pa)
R	Radius (m)
r	Radial distance (m)
U	Velocity magnitude (m/s)
$\vec{U}$	Velocity vector (m/s)
u, v, w	Velocity components (m/s)
V	Volume (m^3)
V_{up}, V_{dn}	Sub-volumes upstream and downstream of the AS (m^3)
X, x, Y, y, Z, z	Cartesian coordinates

x, r, θ	Cylindrical coordinates
α	Angle of attack (rad)
β	Pitch angle of blade (rad)
Γ	Circulation around a segment (m^2/s)
γ	Local circulation (m^2/s)
Δp	Pressure discontinuity (Pa)
$\Delta u, v, w$	Discontinuity of velocity components (m/s)
$\Delta\phi$	Discontinuity of the generic variable
Θ	Segment angle relative to the wind (rad)
θ_1	Angle from random point to segment leading edge (rad)
θ_2	Angle from random point to segment trailing edge (rad)
ρ	Density (kg/m^3)
Φ	Volumetric flux (m^3/s)
ϕ	Generic variable
Ω	Rotational velocity (rad/s)
	Indices
AS	Actuator Surface
av	Average value (associated with the actuator surface)
dn	Downstream of the AS
$face$	Cell face affected by the discontinuity
I	Inertial frame of reference
i	Random node
rel	Relative frame of reference
ref	Reference value

t	Tangent to the Actuator Surface
up	Upstream of the AS
X, x, Y, y, Z, z	Cartesian coordinates
∞	Incoming flow conditions
	Operators
∇	Gradient
$\widetilde{\nabla}$	Corrected gradient
$\nabla\cdot$	Divergence
Φ	Velocity flux
$\widetilde{\Phi}$	Corrected velocity flux
$\otimes$	Tensor product

INTRODUCTION

The need for alternative sources of energy to fossil fuels has caused the renewable energy market to grow exponentially over the past few years. Amongst renewable energy sources, wind power positions itself as one of the most promising ones, as it is plentiful, clean and very little land is needed for the construction of wind turbines. This has led to investment in research activities which has resulted in the construction of more powerful and efficient turbines. The wind energy sector has experienced remarkable growth, from 18 GW at the end of 2000 to over 230 GW installed worldwide in 2011. According to the IEA and CanWEA, installed capacity in Canada totaled over 7.8 GW in 2013, supplying approximately 3% of the country's electricity needs.

Wind turbines extract mechanical energy from the wind and transform it into electrical energy, with the efficiency of the turbines being affected by both the machine and the resource. Knowledge of both is essential for the development and improvement of this technology. From the standpoint of the machine, the construction of precise and economical wind turbine components is dependent on being able to predict the aerodynamic loads that the blade will have to withstand in normal and especially in extreme conditions. Furthermore, demand for this energy source has driven the construction of wind farms in which turbines are grouped in array configurations. This creates an interaction between the turbines and the wakes of other turbines located upstream and downstream.

The study of wind turbine wake aerodynamics arises with the increasing need to understand how the wind affects the turbine and how the turbines affect each other. The nature of the problem is the analysis of the characteristics of the flow around a wind turbine, and of the system of forces generated by the wind over the turbine blades. Over the years many approaches have been taken in an attempt to fully or partially understand wind turbine wakes. Initially, some have collected data from experiments under controlled conditions like wind tunnels. Others have created simplified mathematical models such as vortex methods, kinematics, Blade Element Momentum (BEM), etc., so that by mathematical approximations certain characteristics of the flow can be predicted.

In recent years, computational fluid dynamics (CFD) has become one of the leading tools for understanding wind turbine wakes and their effect on the flow. Several techniques have been developed, including (i) the direct modelling of wind turbines by means of a fitted mesh and (ii) distributed volume forces able to represent fans, propellers, wind turbines, helicopters, amongst other lifting devices. Volume forces can represent lifting devices in the form of the well-known actuator disk, and more recently the actuator line and surface. Although these methods have proven to be effective (especially when studying the far wake characteristics of a turbine), some issues may appear. Problems arise regarding the mesh resolution and blade boundary layers for fitted meshes, as the flow exhibits a viscous behaviour. In the case of volume forces methods, rapid variations or oscillations in flow properties such as velocity and pressure appear. These disturbances appear within the vicinity of the volume force, due to inadequate treatment of flow property discontinuities. Therefore it is a less desirable approach for the representation of the near wake.

As an alternative method for wind turbine wake simulations, Sibuet Watters and Masson (2010) developed a mathematical model that introduced a new kind of technique. This model, based on a method called the Actuator Surface (AS), allows the representation of lifting devices without exhibiting oscillations. The AS is a singular surface, where the action of the modelled device is spread across the surface rather than the volume. The AS carries velocity and pressure discontinuities as well as an attached system of forces. The system of forces is prescribed to represent the action of the wing or blade on the flow, whose magnitude is linked to the wing's circulation properties. The circulation is evaluated using local blade element analysis.

After the AS was successfully implemented in a home code, OpenFOAM (Open Field Operation and Manipulation) was chosen to continue the study of lifting devices using the AS technique. OpenFOAM is an open source program which has a broad and growing community of users. It is also one of the most powerful and flexible CFD programs to date. The open source nature of the program also allows users to modify and implement different applications, such as solvers and boundary conditions. Users can also benefit from modifications imple-

mented by others. This has led to the creation of online cooperation forums, where users from around the world share knowledge and search for solutions to common difficulties.

Objectives and methodology

The main objective of the present work is to study the near and far wake characteristics of a wind turbine modelled with a fully operational AS. Nevertheless, no such model is readily available in OpenFOAM and therefore must be implemented. Before the AS is considered to be fully implemented, the two main characteristics of the AS – pressure and velocity discontinuities – must be implemented and validated. In order to do so, increasingly complex objectives have been adopted:

- To implement a pressure discontinuity normal to the AS, and validate it against an analytical case.
- To implement a velocity discontinuity tangent to the AS, while the pressure discontinuity remains normal to the surface, followed by the validation of the code against an analytical case.
- To apply the AS by modelling a lifting device in translation, such as a wing. To compare the results with an ideal case, whose analytical solution is given by Prandtl's lifting line theory.
- Lastly, to apply the AS by modelling a lifting device in rotation, such as a wind turbine. To compare the results with experimental data.

Thesis organization

The thesis is divided into three main sections: literature review, Actuator Surface model, and application of the AS.

The literature review focused on the components of wind turbine wake and its characteristics. Various data collection methods have been reviewed, including a number of experimental methods. With regard to theoretical methods, various mathematical models have been refer-

enced. Emphasis has been placed on current CFD methodologies for the simulation of wind turbines and their wakes.

Chapter 2 shows the mathematical model of the AS and the modified Navier-Stokes equations. The development of the correction terms and their implementation in OpenFOAM is also portrayed. In this chapter two main sections can be seen. First, the implementation of the pressure discontinuity, which is then compared with two validation cases. The second section concerns the implementation of the velocity discontinuity, which is also compared against a validation case where an analytical solution is available.

Lastly, Chapter 3 discusses applications of the AS. The first application case is the finite wing in uniform translation. The AS will be used to model a wing and its effects on the flow. The results are then compared with Prandtl's lifting line theory. In the final application case, the AS will be used to model the experimental rotor of the Technical University of Delft, whose near wake data are compared with the numerical results.

CHAPTER 1

LITERATURE REVIEW

1.1 Wind turbine wakes

Wind turbines have different effects on the wind flow. These effects are the subject of studies in fluid mechanics, and, more specifically, aerodynamics. These types of studies are highly needed, especially as the wind energy industry continues to grow. Wind turbines extract momentum and energy from the wind flow. This interaction between the turbine and the flow creates a pressure jump and an overall decrease in velocity. As explained in the work of Crespo *et al.* (1999), it can be noticed that as it approaches the turbine, the wind experiences an increase in pressure and a decrease in velocity. Downstream in the immediate vicinity of the turbine there is a non-uniform pressure and axial velocity deficit related to the thrust. Regarding the azimuthal component of velocity, the deficit is related to the torque. This produces a wake downstream of the wind turbine.

This phenomenon is of extreme interest, as in large wind farms the turbine arrangements create an interaction between turbines and their wakes. This will directly affect the overall power production. Whale et al. (1996) mentions that wake characteristics and their relation to the local wind field and topography will affect the energy resource of a site. This relationship will also help determine turbine loading and the minimum inter-turbine spacing in wind farm arrays. Turbine wake is usually divided into two main areas of study: the near wake and the far wake.

The near wake is a complex region that develops immediately after the turbine, usually up to one to three diameters downstream of the turbine. In this region, strong vortical structures called tip vortices can be observed. These structures emanate from the blade tip area describing a helical trajectory. These vortices form a continuous vorticity sheet known as a shear layer. The shear layer increases in size downstream, due to turbulent diffusion. The tip vortices decay within two to three diameters downstream of the rotor. This decay is caused by the ambient

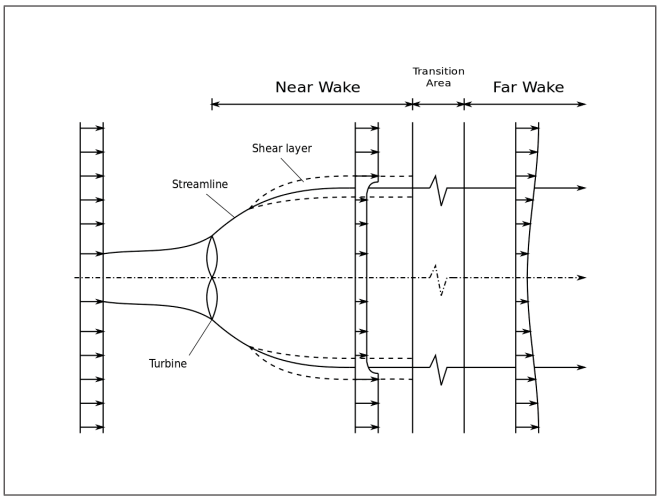

Figure 1.1 Illustration of wind turbine wake.

turbulence. Large-scale turbulence is generated in the shear layer of the wake which spreads into the wake core.

In the near wake area, the axial pressure gradient is important for wake deficit. In this area, peaks in turbulence intensity, velocity gradients and an expansion of the wake width can be observed. Ainslie (1988) states that the wake is also directly affected by the properties of the rotor and the functioning conditions of the wind turbine. Such conditions include: airfoil profile, angle of attack, solidity ratio, aspect ratio, incoming flow characteristics, rotational velocity, blade materials, tower and hub, wind shear, etc. All this will be reflected in the thrust coefficient of the machine.

After a transition area, the far wake is found, usually after five diameters downstream of the turbine. In this area, according to Sanderse (2009) the main focus lies on wake and turbulence modelling, as well as wake interaction and topographic effects. In the far wake region, the wake is completely developed and there is little influence from the rotor. The wake is described by a reduced velocity field and an increase in turbulence. Convection and turbulence diffusion are the two main effects that determine the flow conditions. Turbulence is the dominating aspect of the far wake. According to the work of Sanderse *et al.* (2011), three sources of turbulence can be identified:

- The atmospheric turbulence due to thermal effects and surface roughness.
- The mechanical turbulence due to the effects of the blades and tower.
- The wake turbulence due to tip vortex brake-down.

In the far wake, if topographical effects are disregarded, the velocity deficit and turbulence intensity properties can be assumed to be self-similar and axisymmetric. This assumption is the basis of many wind turbine wake models. The far wake of wind turbines is studied by single wake models as well as models representing several wakes, also known as wind farm models. Wind farm models are concerned with the interaction between wind turbines and their wakes. During this interaction, not only will the velocity deficit affect the ability of downstream turbines to extract momentum from the flow, but also the increased turbulence will affect the performance of other turbines.

In order to understand the wake and its characteristics data must be collected. This can be achieved either by experimental investigation or theoretical calculation.

1.2 Experiments

Extraction of data from experiments by actual measurements comes as a natural first option, as is often the most reliable information source on a physical process. Different organizations and universities have collected experimental data. These data are not only useful for understanding the nature of the wake but are also important to validate numerical models. The data can be collected from field experiments but are most commonly gathered in wind tunnels where experimental conditions are easier to control. As seen in the work of Vermeer *et al.* (2003), experimental techniques include hot wires, static pressure, hot film anemometry, laser Doppler velocimetry (LDV), particle image velocimetry (PIV), and pressure tabs.

The standard modern turbine consists of three blades, and the most promising results come from experiments of full-scale turbines in wind tunnels. These experiments are able to predict how identical copies of the equipment would behave under the same conditions. Because of their size, the cost of conducting such experiments is prohibitive. Consequently, most experi-

ments are conducted with scale models. Further, many of these experiments are conducted with two- or even one-blade turbines, which also provide valuable information on wake properties.

Some of the best known experimental wind turbines used in the validation of mathematical models are described below.

The MEXICO project, or Model Experiments in Controlled Conditions, was a project using a three-bladed turbine and designed to create a detailed database of wind turbine aerodynamics and loading. The objective was for these data to be used for model validation and improvement. The main focus was the study of the near wake using PIV. This project involved partners such as the Energy Research Centre of the Netherlands (ECN), the Technical University of Delft, Risø National Laboratory, Technical University of Denmark, amongst others. Reports of the MEXICO project detailing the experiments and results can be found in the work of Snel *et al.* (2007) and Schepers and Snel (2007). Breton *et al.* (2010) applied the Actuator Surface technique to the MEXICO turbine with remarkable results.

The TUDelft experimental rotor is the two-bladed experimental rotor of the Technical University of Delft, and is also one of the most popular rotors for the study and model validation of near wake aerodynamics. The experimental measurements were taken using hot film anemometry. Details about the experiment and the results can be found in the work of Haans *et al.* (2008) and Sant (2007). Sibuet Watters and Masson (2010) applied the Actuator Surface concept to the TUDelft rotor, which has also been used as an application case in the present work.

The NREL phase VI two-bladed experimental rotor is another widely used rotor for code validation and improvement. The data collection was performed at the NASA Ames Research Center wind tunnel using pressure tabs. The results have been referenced by many authors including Duque *et al.* (2003), Sørensen *et al.* (2002), Johansen *et al.* (2002), amongst others. The Actuator Surface was also applied to this turbine by Watters and Masson (2007).

Collecting data from experiments in general poses some disadvantages because of its cost, limited range of data available, the inability to control every variable of the experiment, scaling problems, and measurement equipment which is not infallible. These disadvantages become

more evident in field experiments. This has led to new alternatives and techniques to be developed in order to simulate turbines and other lifting devices (helicopters, propellers, wings, etc.), and approximate their behaviour.

1.3 Computational fluid dynamics

Computational fluid dynamics (CFD), is the branch of fluid dynamics that uses numerical methods and algorithms to analyze a fluid flow. In the case of the wind industry, it is usually based on the Navier-Stokes equations. Within CFD different methods can be found to simulate wind turbines and their effect on the flow.

1.3.1 Direct modelling

Direct modelling consists of constructing a body fitted mesh based on the exact geometry of the blade. The main objective is to model the exact flow around the blade and consequently analyze the wake that develops downstream of the turbine. Even if the exact modelling of the turbine represents an excellent first option, many problems arise when using this technique. It is computationally expensive, and an adequate choice of the blade boundary layer is not evident. The flow around the blade may be compressible while a remaining incompressible behaviour may be observed in the rest of the domain. Finding an adequate mesh refinement to model the blades can also be an issue.

1.3.2 Volume forces

As an alternative to direct modelling, the representation of the rotor is done by volume forces. This representation helps avoid the modelling of the blade boundary layers, saves computational costs and eases the grid generation.

When using volume forces, the modelled lifting device is represented by a force opposed to the flow resulting in a momentum sink. In this kind of treatment the force is explicitly introduced

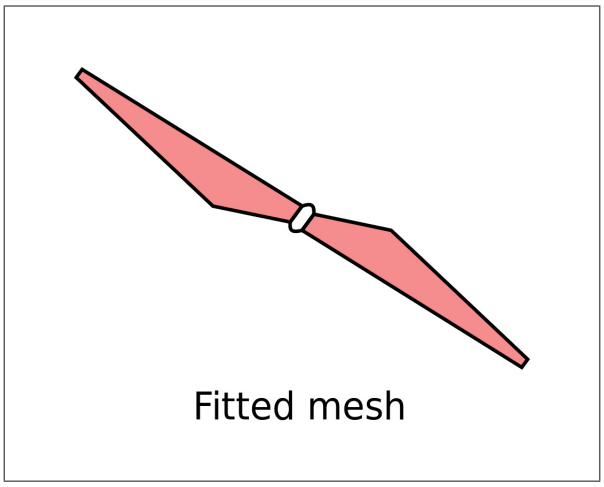

Figure 1.2 Illustration of direct modelling.

in the flow by modifying the Navier-Stokes equations. The momentum sink induces a pressure jump and a decrease in velocity.

Even with its significant advantages, this kind of technique of flow modelling usually induces disturbances or oscillations in the flow within the vicinity of the volume force. It therefore generates an inadequate representation of the near wake characteristics of the wind turbine. The model nevertheless functions well in the far wake region, where the flow characteristics are less influenced by the turbine geometry.

One way to avoid spatial oscillations caused by the volume forces is to use distributions. This however is not an adequate representation of the sudden change in pressure caused by the turbine. Progress has been made in this area by Rethore and Sørensen (2008), where the pressure jump was made in 3 nodes.

1.3.2.1 Actuator disk

Volume forces can take different shapes to represent the turbine. The actuator disk is arguably the most common of these. In this technique, an infinite number of blades is assumed in the rotor swept area. The forces are present over the disk surface by uniform and non-uniform distributions.

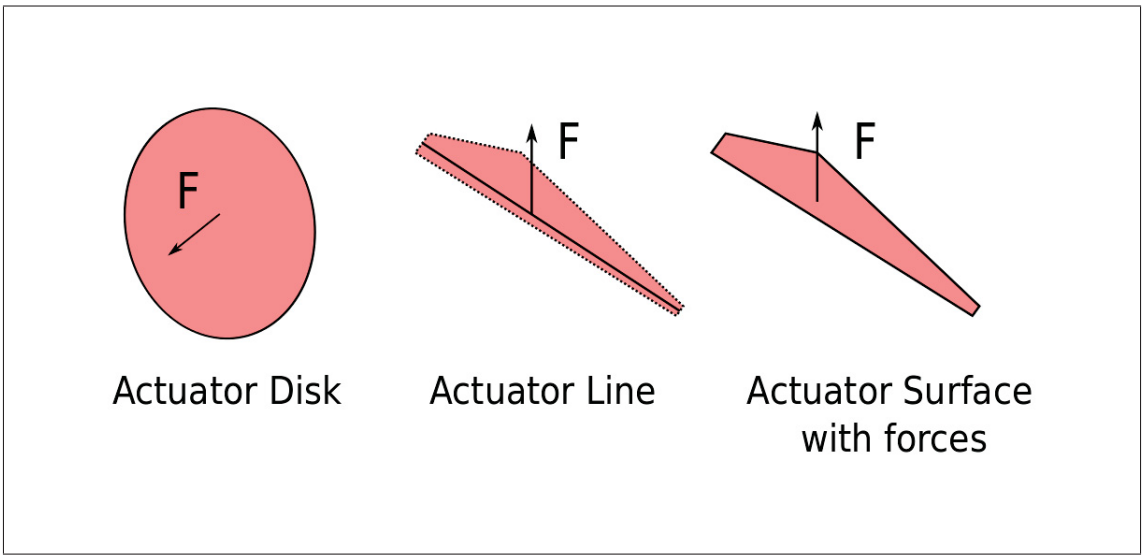

Figure 1.3 Illustration of the actuator disk, actuator line and actuator surface.

In the case of a uniform distribution the force f is usually a function of the incoming reference velocity U_{ref} and the thrust coefficient C_T:

$$f = \frac{1}{2}\rho U_{ref}^2 C_T \tag{1.1}$$

Where ρ represents the density of the air. When the disk is perpendicular to the incoming flow and the flow is undisturbed the reference velocity is equal to the incoming flow velocity U_∞. In the case of a disturbed flow, techniques to determine the incoming reference velocity can be found in the work of Meyers and Meneveau (2010), Calaf *et al.* (2010) and Prospathopoulos *et al.* (2009).

When speaking of non-uniform distributions the value of the radial component of the force varies but usually remains uniform in the azimuthal direction. Discrete distributions can be introduced such as the Goldstein distribution for rotor and propellers. Some distributions may calculate forces by using the local velocity components, while others can take experimental data similar to the blade element momentum (BEM) method.

Apart from the axial force representing the thrust, forces tangential to the disk can be prescribed to represent the effects of rotation. As shown in the work of Porté-Agel *et al.* (2010), including the effects of rotation improves the prediction of velocity and turbulence intensity.

1.3.2.2 Actuator line

In the actuator line technique, the blade is discretized by a finite number of nodes in a straight line. This line may appear static if using a rotating frame of reference, or it may rotate itself. In both cases the forces, usually calculated form airfoil data, are not averaged as in the case of a non-uniform actuator disk but rather depend on the rotational velocity. In most actuator line models, information about the drag and lift of the blade is required. The effects of Coriolis and centrifugal forces must be taken into account.

According to Sanderse *et al.* (2011), one of the main differences with the actuator disk model is that the actuator disk effect on the flow is shed continuously, like a vortex sheet. In the case of an actuator line the apparition of tip vortex structures is observed.

1.3.2.3 Actuator Surface

With regard to volume forces, the Actuator Surface comes as an extension of the actuator line technique. In this case the blade is represented by a planar surface that may or may not take a similar shape to that of the blade. The volume forces are distributed over the surface. The forces may be calculated using the lift and drag coefficients as in the work of Dobrev *et al.* (2007), or may even take pressure and skin friction distributions as found in the work of Shen *et al.* (2007).

1.3.3 Actuator Surface using direct pressure and velocity discontinuities

The Actuator Surface (AS) technique concerns itself with the insertion of direct pressure and velocity discontinuities in the flow. This is achieved by means of explicit correction terms.

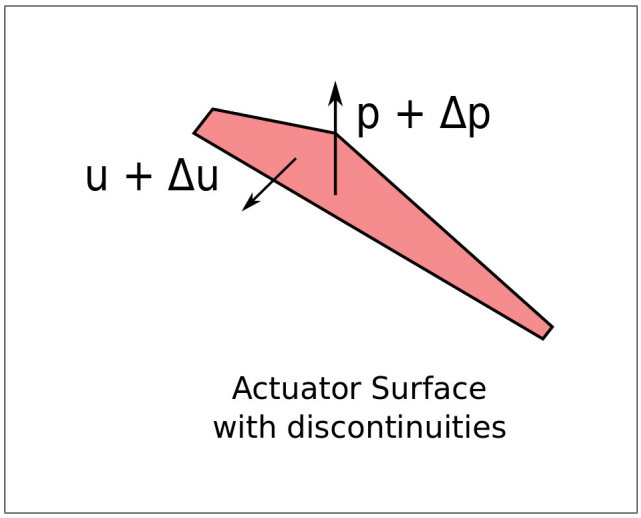

Figure 1.4 Illustration of the Actuator Surface with discontinuities

These terms are inserted in the incompressible Navier-Stokes equations. In the work of Masson *et al.* (2001) the following theorem can be found:

$$\int_A \nabla p dA = \int_{V_{up}} \nabla p dV + \int_{V_{dn}} \nabla p dV + \int_{A_{AS}} \nabla (p_{up} - p_{dn}) dA_I \tag{1.2}$$

Where A is the outer area of a control volume, A_{AS} is the AS area, p is the pressure, p_{up} and p_{dn} are the values of pressure upstream and downstream of the AS, and V_{dn} and V_{up} are the sub-volumes within the immediate vicinity of the AS. This is shown in Figure 1.5. The condition imposed by the divergence theorem allows for a sudden change in pressure (pressure jump) to occur in one node.

The AS also acts as a vortex sheet closely resembling the lifting line theory and vortex methods. The AS is only applied on inviscid flows. In this technique the circulation is calculated by using experimental airfoil lift data and the discontinuities are distributed using a parabolic distribution. The AS can take the same shapes as the volume forces but the main difference

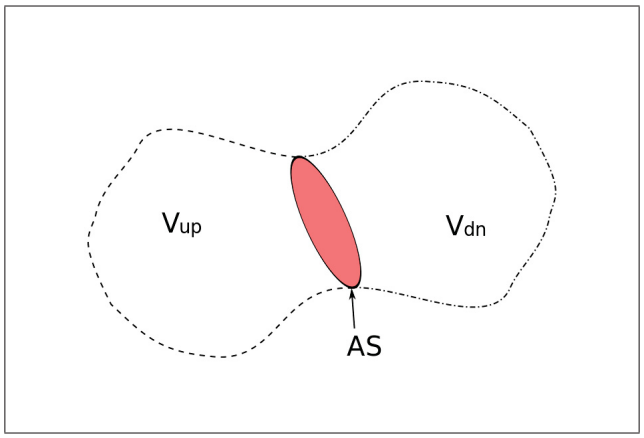

Figure 1.5 Analysis of the representative preamble surface action.

lies in the near wake results. No spatial oscillations are observed in the AS model near the area of discontinuity, thus making it an appropriate choice for near and far wake analysis.

The development and validation of this method can be found in the work of Sibuet Watters and Masson (2010) and Sibuet Watters (2009).

1.4 Other methods

Other methods to predict wind turbine and wake behaviour include kinematics, the momentum method, blade element momentum (BEM), and vortex methods.

Kinematics is an analytical method based on the self-similar nature of the far wake. As mentioned in the work of Crespo *et al.* (1999), velocity profiles are usually obtained from theoretical and experimental data. Different profiles are assigned to near and far wake. By means of global momentum conservation, results of velocity deficit and turbulence intensity are found from the thrust coefficient of the machine.

The momentum method is a classical method in which the rotor is modelled by an actuator disk. Snel (1998) states that the momentum and energy flow are enclosed in a control volume composed of the stream tube that encloses the rotor. This method was applied successfully for a uniformly loaded actuator disk in an axisymmetric steady flow. This gave rise to expressions

such as the well-known Betz limit, where the power coefficient C_P equals 16/17 of the wind's kinetic energy. This method was further developed to include the rotational effects of the turbine.

Blade element momentum (BEM) is another widely used method in which the blade is divided into N number of elements, and no radial flow conditions are imposed. This combines global momentum balance with 2D blade elements analysis. This is based on experimental data from lift and drag coefficients, which come as a consequence of the angle of attack. It also calculates the aerodynamic blade characteristics.

Vortex methods assume an inviscid incompressible flow. In this kind of method the blade strength is determined by its bound circulation. In the work of Hansen *et al.* (2006), two vortex methods are seen: the vortex-lattice method, in which the blade is represented by vorticity sheets or lines, a nd vortex-panels, in which the blade geometry is taken into account. As in BEM, vortex methods may take experimental data.

CHAPTER 2

ACTUATOR SURFACE MODEL

The present chapter is devoted to the implementation and validation of the Actuator Surface. The AS will be implemented following the principles of incompressible fluid dynamics applied to inviscid flows. To achieve the full implementation of the AS both pressure and velocity discontinuities must be inserted in the flow.

The mathematical model presented is based on the work of Sibuet Watters and Masson (2010). The implementation has been done using the open source software OpenFOAM. To implement the AS, explicit correction terms are inserted in the Navier-Stokes equations, which are also presented. Specific validation cases have been put into place where an exact analytical solution can be found. The results are then compared with the analytical solution.

2.1 Mathematical Aspects of the AS

For steady incompressible non-viscous flow conditions, the integral Navier-Stokes equations can be written as:

$$\int_A \nabla \cdot (\vec{U}) dA = 0 \tag{2.1}$$

$$\int_A \nabla \cdot (\vec{U} \otimes \vec{U}) dA = -\int_A \nabla p dA \tag{2.2}$$

Where $\vec{U}$ is the velocity vector, p refers to the pressure and A denotes the outer surface of a control volume V. Most CFD methods use the incompressible form of the momentum equations. When using these equations, the flow in and around the wake has a similar behaviour. Only in the case of direct modelling, compressibility effects might arise around the blade tips. In the case of the AS method, the viscous terms have also been removed, as required by the model.

2.1.1 Definition of the AS

The Actuator Surface is defined as a porous surface carrying pressure and velocity discontinuities, normal and tangential to the surface, respectively. The AS may also be regarded as an infinitely thin vortex sheet. This vortex sheet generates lift forces linked to its circulation properties.

A Cartesian system of coordinates (X,Y, Z) is defined. In this system the surface is positioned on the XY plane. Therefore all discontinuities will occur in the Z direction. Figure 2.1 illustrates the nature of the AS.

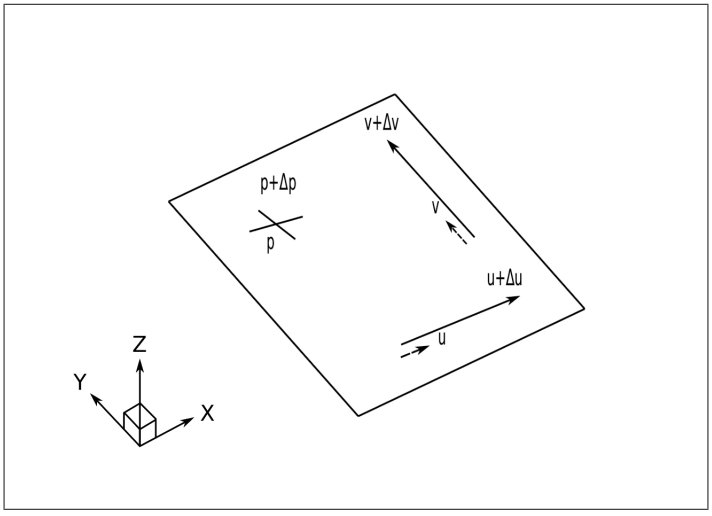

Figure 2.1 Simple Actuator Surface.

A variable ϕ is defined. This variable can take the form of the components of $\vec{U}$ (u, v, w) and p. The discontinuities of ϕ can be measured as follows:

$$\Delta\phi = \phi_{Z_{P+}} - \phi_{Z_{P-}} \tag{2.3}$$

In order to be able to model a lifting device correctly the AS must guarantee:

- Zero net vorticity flux for any control volume that intersects with the AS.

- Zero exchange of energy between the AS and the flow relative to the AS.

In order to satisfy these requirements two conditions are applied. First, the following constraint is applied to the velocity discontinuities to ensure the zero flux condition:

$$\frac{\partial \Delta v}{\partial x} - \frac{\partial \Delta u}{\partial y} = 0 \tag{2.4}$$

The second condition states that the system of forces attached to the AS (all in surface area per unit) must not exert mechanical work on the flow relative to the AS. Taking this condition into account the forces in all three directions can be expressed as follows:

$$f_x = \rho w_{av} \Delta u \qquad f_y = \rho w_{av} \Delta v \tag{2.5}$$

$$f_z = -\rho(u_{av} \Delta u + v_{av} \Delta v) \tag{2.6}$$

In the calculation of the forces attached to the AS the suffix av appears. av indicates the average value of velocity on both sides of the AS:

$$u_{av} = \frac{u_{Z_{P-}} + u_{Z_{P+}}}{2} = u_{Z_{P-}} + \frac{\Delta u}{2} = u_{Z_{P+}} - \frac{\Delta u}{2} \tag{2.7}$$

In the case of the normal component of velocity w_{av}, only the first part of the formula applies. This happens because the velocity field is continuous normal to the surface. Hence Δw does not exist. The nature of the Actuator Surface will ensure that the normal component of the

force balances itself with a pressure jump that naturally arises. Therefore it can be deduced that: $f_z = \Delta p$.

Once the forces involved in the AS have been calculated they must be introduced in the Navier-Stokes equations. A detailed explanation of the theory behind the modification of the Navier-Stokes equations can be found in the work of Masson *et al.* (2001). These modifications lead to the following expression:

$$\int_A \nabla \cdot (\vec{U} \otimes \vec{U}) dA = -\int_{V_{up}} \nabla p dV - \int_{V_{dn}} \nabla p dV + \int_{A_{AS}} f_t dA \tag{2.8}$$

In which f_t represents the tangent force associated with the AS. In these modified equations the pressure discontinuity has cancelled itself out with the normal component of the force. V_{up} and V_{dn} represent the sub-volumes within the immediate vicinity of the AS.

2.2 Numerical Implementation of the AS in OpenFOAM

As mentioned in the previous section the open source software OpenFOAM has been used in the present work. Following the method proposed by Sibuet Watters and Masson (2010), the SIMPLE algorithm has been chosen to perform all simulations.

OpenFOAM uses a finite volume method (FVM). FVM is a discretization scheme for a flow domain in which a set of transport equations apply. Once the equations are integrated, usually using the midpoint rule, the result is a finite number of linear equations. These equations can be solved using matrix methods. In FVM the domain is divided into a finite number of control volumes or cells. In OpenFOAM the nodes are located in the middle of the cells of the discretized domain.

In incompressible methods if pressure and velocity are discretized at the same point, also known as a collocated grid, a decoupling between pressure and velocity may occur. This results in spatial oscillations. To avoid this kind of problem some numerical methods calculate

the pressure values at the cell centre, while keeping the velocity values at the cell faces. This is known as a staggered grid (Figure 2.2).

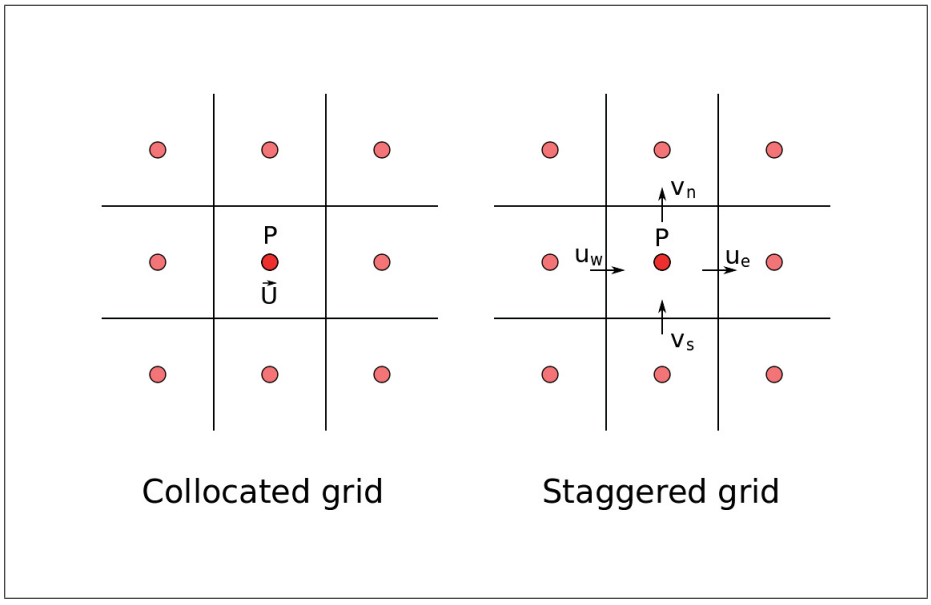

Figure 2.2 Example of collocated and staggered grids.

In the case of OpenFOAM, a collocated grid is always employed. A special treatment of pressure to avoid decoupling is used. This treatment is known as the Rhie-Chow correction algorithm. Kärrholm (2008) explains how the Rhie-Chow algorithm is implemented in OpenFOAM.

A detailed explanation of the implementation of the SIMPLE algorithm in OpenFOAM can be found on the website OpenFOAMWiki. [1]

OpenFOAM comes with a set of boundary conditions unique to the program. Table 2.1 describes the boundary conditions which will be used in the for thcoming simulations. Further details about the boundary conditions available in OpenFOAM can be found on the website of the OpenFOAM foundation. [2]

[1] For further details please refer to: http://OpenFOAMwiki.net/index.php/OpenFOAM_guide/The_SIMPLE_algorithm_in_OpenFOAM

[2] Detailed guide on the boundary conditions is available in OpenFOAM: http://www.OpenFOAM.org/docs/user/boundaries.php

Table 2.1 OpenFOAM boundary conditions.

Boundary conditions	
zeroGradient	The gradient of a variable ϕ normal to the boundary condition is set to zero.
slip	Sets the boundary to zeroGradient if the variable is a scalar. If the variable ϕ is a vector, the normal component is set to zero, while tangential components are set to zeroGradient.
fixedValue	A uniform or non-uniform value must be prescribed for the variable ϕ.
cyclic	Allows two patches to behave as if they were physically connected. Mapping the values of the master patch into the slave patch. With the help of the rotational option the values may be mapped taking into account a centre of rotation.

2.2.1 Pressure Discontinuity

The first step towards the complete implementation of the AS is the pressure discontinuity. For the time being only the thrust will be modelled (the tangent forces are not taken into account). For simplicity a uniform pressure discontinuity expressed in terms of the thrust coefficient C_T and the magnitude of the incoming flow U_∞ is prescribed during the implementation. The discontinuity is given by:

$$\Delta p = \frac{1}{2}\rho U_\infty^2 C_T \tag{2.9}$$

2.2.1.1 Numerical Model

An implementation of the SIMPLE algorithm is present in OpenFOAM. This algorithm is modified by the introduction of explicit correction terms. The correction terms must be inserted in the momentum equation carrying the pressure discontinuity. These correction terms have to be included everywhere the pressure is present.

In the case of the OpenFOAM implementation of the SIMPLE algorithm the pressure is present in:

- The pressure gradient.
- The pressure surface gradient.
- The divergence of the pressure gradient (present in the pressure equation).

In the present implementation a uniform grid (shown in Figure 2.3) is assumed. This is done to illustrate the introduction of the correction terms that will generate the pressure jump. In this example the AS is located at node i.

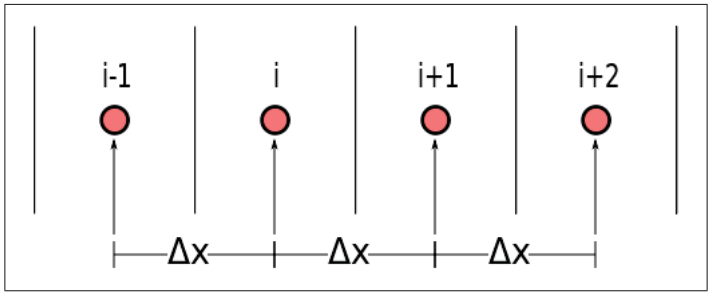

Figure 2.3 Nodes around the AS.

The pressure discontinuity occurs in the direction of the incoming flow U_∞ (X direction in Cartesian coordinates). The pressure gradient at the node i is calculated in the X direction. A second order central differencing scheme is used:

$$\nabla p_i = \frac{\frac{p_{i+1}-p_i}{\Delta x} + \frac{p_i-p_{i-1}}{\Delta x}}{2} = \frac{p_{i+1} - p_{i-1}}{2\Delta x} \tag{2.10}$$

In the proposed method the insertion of explicit correction terms will allow two values of pressure at the node i. The pressures $p_{i_{up}}$ and $p_{i_{dn}}$ are defined by the following expressions:

$$p_{i_{up}} = p_i \tag{2.11}$$

$$p_{i_{dn}} = p_{i_{up}} - \Delta p \tag{2.12}$$

As can be observed $p_{i_{up}}$ is the undisturbed value of pressure at node i, while $p_{i_{dn}}$ is affected by the discontinuity. It should be noted that $p_{i_{dn}}$ is taken into account for all calculations downstream of the AS. Therefore the corrected pressure gradient at node i can be expressed as:

$$\widetilde{\nabla} p_i = \frac{\frac{p_{i+1} - p_{i_{dn}}}{\Delta x} + \frac{p_{i_{up}} + - p_{i-1}}{\Delta x}}{2} = \frac{\frac{p_{i+1} - (p_{i_{up}} - \Delta p)}{\Delta x} + \frac{p_{i_{up}} + - p_{i-1}}{\Delta x}}{2} = \frac{p_{i+1} - p_{i-1}}{2\Delta x} + \frac{\Delta p}{2\Delta x} \tag{2.13}$$

Where $i-1$ and $i+1$ are the neighbouring nodes upstream and downstream of the AS. As mentioned previously, the action of the pressure discontinuity affects the value of pressure downstream of the AS. When the pressure gradient is calculated in the rest of the domain the term $p_{i_{dn}}$ reappears. This occurs during the calculation of the pressure gradient at node $i+1$. Consequently the gradient at node $i+1$ must be corrected as well:

$$\widetilde{\nabla} p_{i+1} = \frac{p_{i+2} - p_i}{2\Delta x} + \frac{\Delta p}{2\Delta x} \tag{2.14}$$

As a consequence of the implementation of the Rhie-Chow algorithm, a pressure equation is created. This happens by applying the divergence operator on the discretized momentum equation. Correction terms are therefore calculated in a similar fashion and included in the divergence of the pressure gradient. This procedure applies for both nodes i and $i+1$:

$$\nabla \cdot (\frac{1}{a_i^U} \widetilde{\nabla} p_i) = \frac{1}{a_i^U} (\frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2}) + \frac{1}{a_i^U} \frac{\Delta p}{\Delta x^2} \tag{2.15}$$

$$\nabla \cdot (\frac{1}{a_{i+1}^U} \widetilde{\nabla} p_{i+1}) = \frac{1}{a_{i+1}^U} (\frac{p_{i+2} - 2p_{i+1} + p_i}{\Delta x^2}) - \frac{1}{a_{i+1}^U} \frac{\Delta p}{\Delta x^2} \tag{2.16}$$

As part of the pressure correction algorithm the face fluxes must receive the influence of the pressure surface gradient. In the case of the surface gradient only the surface within the immediate proximity downstream of the AS is affected by the discontinuity:

$$\widetilde{\nabla} p_{face} = \frac{p_{i+1} - p_i}{\Delta x} + \frac{\Delta p}{\Delta x} \tag{2.17}$$

Where the suffix $face$ refers to the face between nodes i and $i + 1$. Once the correction terms are in place the performance of the code is tested with validation cases. The modified solver is called simpleFoamPJ[3], which stands for simpleFoam Pressure Jump.

To validate the implementation of the pressure discontinuity two cases have been established. The infinite plane and the case of a lightly loaded actuator disk. The results of both cases have been compared with their respective analytical solutions. The main objective of this validation is to test the accuracy of the model, as well as to verify if the conditions of a sudden change in pressure and a continuous velocity field are respected.

2.2.1.2 Validation Case: Infinite plane

The first case of study is the case of the infinite plane. Figure 2.4 illustrates the nature of the problem. In this case a uniform channel flow encounters a plane that divides the domain into two halves. The plane is located in the middle of the domain and carries a uniform pressure discontinuity. It is important to note that OpenFOAM requires boundary conditions to be set for the pressure field. The boundary conditions as well as the discretization schemes can be seen in Table 2.2. In the case of the velocity field, both first order and second order schemes were used during the simulations. With regard to the domain considerations, this case has been

[3]The source code plus an application case for OpenFOAM(2.2.1) can be found in the attached DVD-R and a tutorial is available in Appendix I.

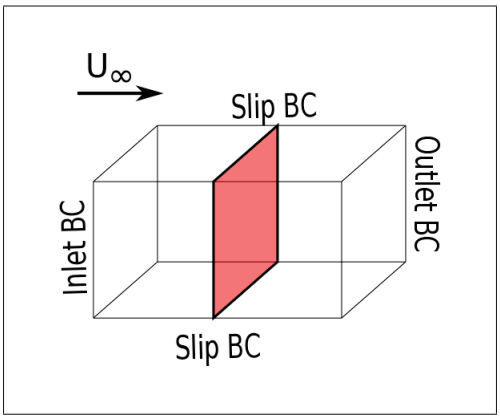

Figure 2.4 Illustration of the infinite plane case.

thoroughly studied with several domain and cell sizes to discretize both the domain and the AS.

Table 2.2 Boundary conditions and schemes for the infinite plane case.

	Walls	**Inlet**	**Outlet**	**Scheme**
Velocity	slip	fixedValue	zeroGradient	first & second order
Pressure	zeroGradient	zeroGradient	fixedValue	second order

This arrangement of boundary conditions ensures that the flow will remain unidirectional. Only a sudden change in pressure is expected. The change in pressure should be decreasing from one side to the other of the discretized AS.

2.2.1.3 Results: Infinite plane

The results of the simulation performed with the modified solver are shown in Figures 2.5 and 2.6. In Figure 2.5 the normalized pressure can be observed, while in Figure 2.6 the normalized velocity is observed. In both cases the results are plotted against the normalized distance. Where the distance x is divided by the AS side B. The AS is located at 0. As expected, a sudden change in pressure is observed at the AS, while no change occurs in the velocity field. The numerical result closely matches the analytical solution. The results do not present anomalies such as oscillations within proximity of the AS, as simulations using volume forces generally

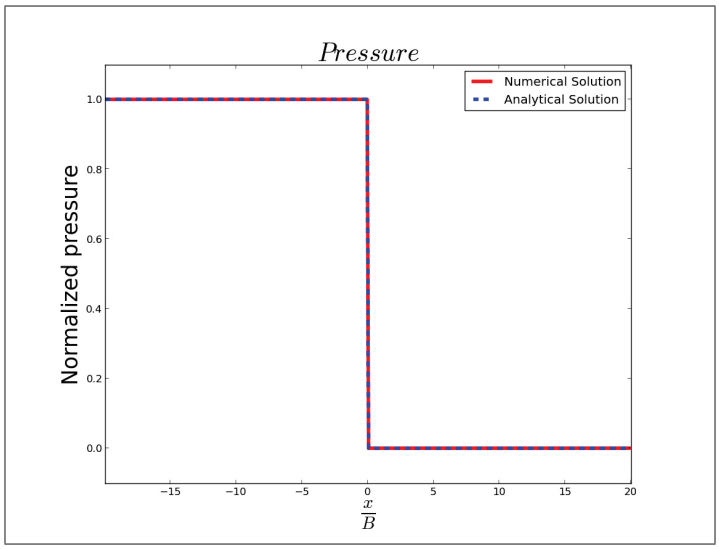

Figure 2.5 Infinite plane case: pressure compared with analytical solution.

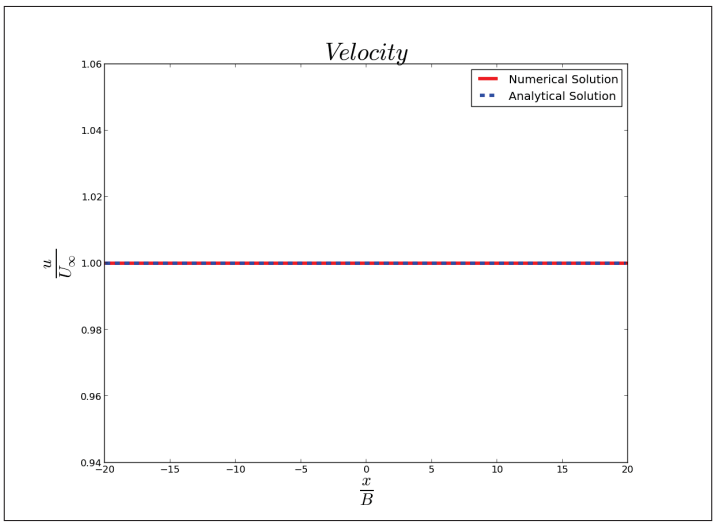

Figure 2.6 Infinite plane case: velocity compared with analytical solution.

do. The two pressure values held in the AS nodes can be observed, in agreement with the theory presented in the previous sections.

Two limitations are noted in the current implementation, the first being that the model needs a uniform mesh in the direction of the pressure discontinuity around the AS, consisting of one node upstream of the AS, and two nodes downstream, for a total of four uniform cells.

The second limitation is related to the corrections terms, which are conceived for a second order central differencing pressure scheme. This is the only possible scheme that can be used with current implementation of the AS.

2.2.1.4 Validation Case: Lightly loaded actuator disk

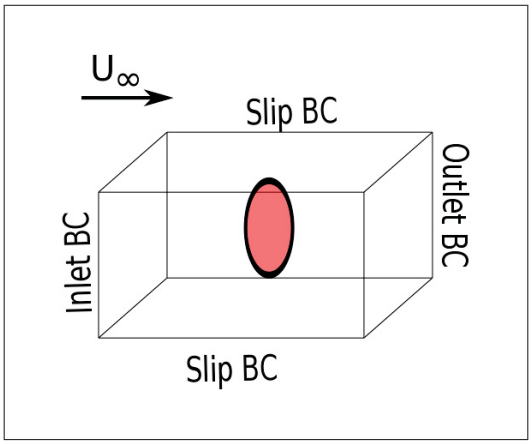

Figure 2.7 Illustration of the lightly loaded actuator disk case.

The second validation case is that of the well-known actuator disk problem. It is interesting to analyze this case as an actuator disk is a special form of the AS. This very case has been implemented and analyzed by several authors. A full analytical solution for a lightly loaded actuator case was derived by Koning (1935) in cylindrical coordinates.

$$p(x,r,\theta,\Delta p,R) = \frac{\Delta p}{4\pi}\int_0^R\int_0^{2\pi}\frac{r^{'}x}{[r'^2+r^2+x^2-2r'rcos(\theta'-\theta)]^{3/2}}dr'd\theta \tag{2.18}$$

$$u(x,r,\theta,\Delta p,R) = U_\infty - \frac{p(x,r,\theta,\Delta p,R)}{\rho U_\infty} - \underbrace{\frac{\Delta p}{\rho u_\infty}}_{\text{only in the wake}} \tag{2.19}$$

Where U_∞ is the magnitude of incoming velocity. It is important to note that the analytical solution is only valid for a lightly loaded actuator disk. In this case a thrust coefficient (C_T) of 0.01 has been chosen. This same value of C_T can be found in the work of Rethore and Sørensen (2008). The boundary conditions and schemes used in the case are similar to those used for the infinite plane and can be seen in Table 2.3.

Table 2.3 Boundary conditions and schemes for the actuator disk case.

	Walls	**Inlet**	**Outlet**	**Scheme**
Velocity	slip	fixedValue	zeroGradient	first & second order
Pressure	zeroGradient	zeroGradient	fixedValue	second order

In the case of the lightly loaded actuator disk, first and second order convection schemes were also used. No noticeable difference in the results was found. Regarding the code, not only must the performance of the solver be addressed, but its flexibility must also be tested. In the infinite plane case the AS occupies the entire plane. The question is whether the four node constraint applies for the entire plane or only for the nodes where the AS is defined.

2.2.1.5 Results: Lightly loaded actuator disk

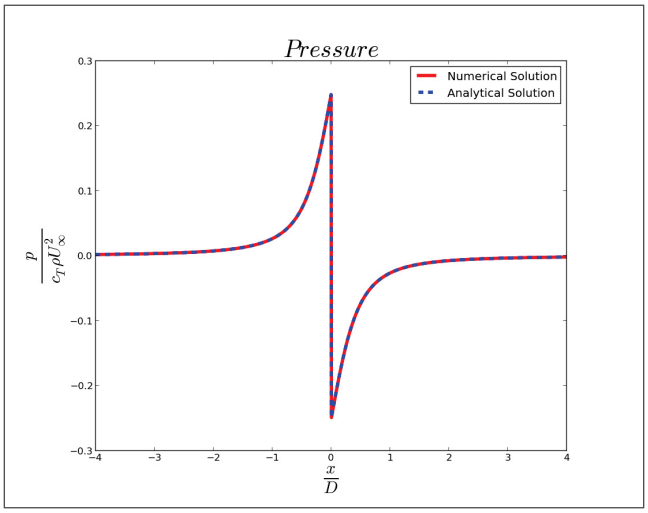

Figure 2.8 Lightly loaded actuator disk case: pressure compared with analytical solution.

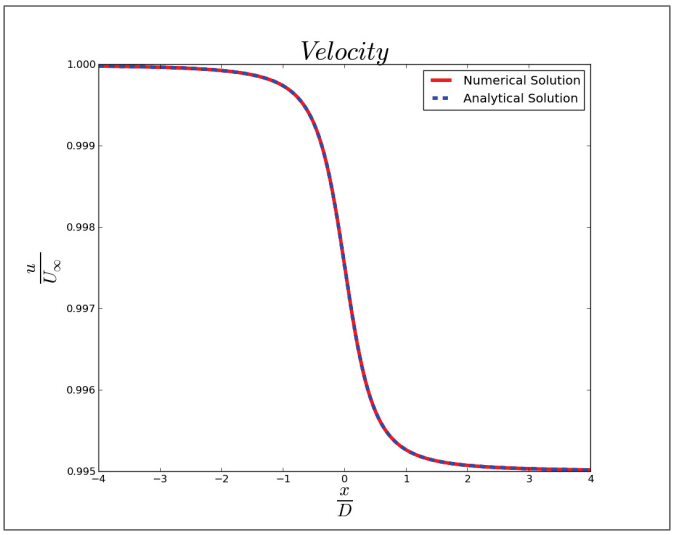

Figure 2.9 Lightly loaded actuator disk case: velocity compared with analytical solution.

A similar behaviour to the infinite plane case is observed in Figures 2.8 and 2.9. A sudden change in pressure normal to the actuator disk is found. The velocity field remains continuous throughout the AS. This is in excellent agreement with the analytical solution.

Figure 2.10 Lightly loaded actuator disk case: wake development.

Figure 2.10 shows the effect of the AS on the flow. A decrease in the velocity field can be clearly observed. The creation of the wake downstream of the AS is also portrayed. The next question that was faced was that of mesh flexibility. Several cases have been analyzed using uniform to extreme mesh gradings, as well as different domain shapes and sizes. This was done for both the domain and the actuator disk. It was concluded that only the nodes belonging to the AS require the four-node uniform mesh. Once again, this must be done in the direction of the discontinuity (normal to the surface). The mesh belonging to the rest of the domain can be handled as it is deemed more appropriate.

2.2.2 Velocity Discontinuity

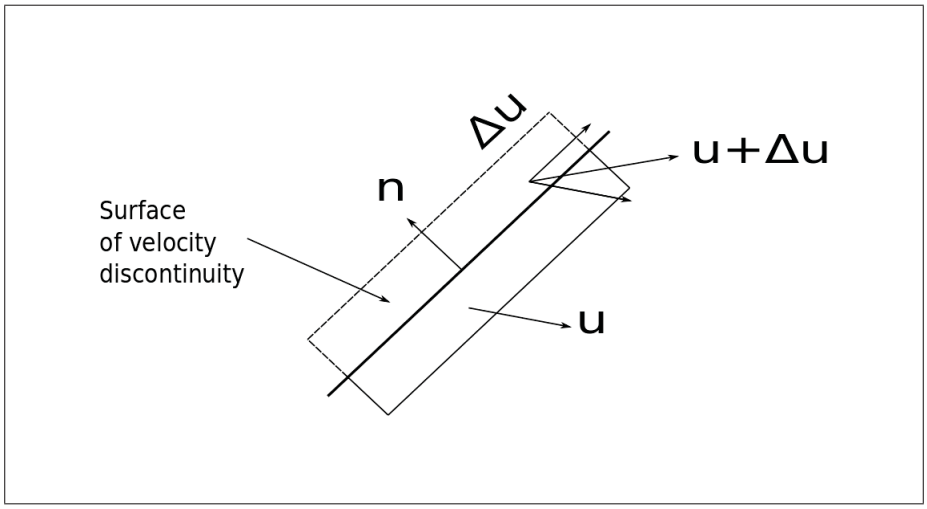

Figure 2.11 Velocity discontinuity.

The next step to obtain a fully operational AS is the implementation of the velocity discontinuity. In this case the pressure discontinuity remains normal to the AS. The velocity discontinuities appear tangent to the surface, and they are distributed over a surface of zero thickness. If Figure 2.11 is analyzed, it can be observed that in the immediate vicinity of the surface the velocity $u + \Delta u$ is on the upper side of the surface of discontinuity. The u velocity is located on the lower side. For simplicity, in order to calculate the velocity discontinuities the total circulation Γ must be prescribed. For the validation case, a uniform distribution of discontinuity

along the segment chord c is assumed. The uniform distribution of discontinuity is shown in Equation 2.20.

$$\Delta u = \Gamma / c \tag{2.20}$$

2.2.2.1 Numerical Model

Once the total circulation is prescribed and the discontinuities have been determined, implementation of the velocity discontinuity is straightforward. The tangent forces are calculated using the method presented in the mathematical model. The calculation of these forces can be seen in Equations 2.5 and 2.6. The forces must be explicitly introduced in the Navier-Stokes equations as shown in Equation 2.8. In the same manner the velocity fluxes Φ (calculated at the cell faces) must account for the velocity discontinuities. The flux in the X direction can be expressed as:

$$\Phi = (H \frac{1}{a^U})_{face} A \tag{2.21}$$

Equation 2.21 shows how Φ is simply the product of two interpolated coefficients: the H coefficient obtained from the momentum matrix (H includes the source terms and neighbour cell contributions) and $\frac{1}{a^U}$ coefficient which corresponds to the matrix diagonal divided by the cell volume. The product of the two coefficients is then multiplied by the face area. An explanation regarding both coefficients is found in the work of Jasak (1996).

To introduce the flux corrections, a similar procedure as that of the mathematical model is followed where the tangent forces are calculated at the cell faces:

$$f_{x\,face} = \rho w_{av\,face} \Delta u_{face} \qquad f_{y\,face} = \rho w_{av\,face} \Delta v_{face} \tag{2.22}$$

Once the tangent forces at the cell faces are calculated, the corrected flux in OpenFOAM is equal to:

$$\widetilde{\Phi} = \Phi + (\frac{1}{a^U} f_t)_{face} A_{AS} \tag{2.23}$$

In this case the product of the face tangent forces and the interpolated coefficient $\frac{1}{a^U}$ is multiplied by the face area. Once the changes are put into place, the new application is called simpleFoamVD[4]. The name of the solver stands for simpleFoam Velocity Discontinuity.

2.2.2.2 Validation Case: 2D Velocity discontinuity segment with uniform distribution

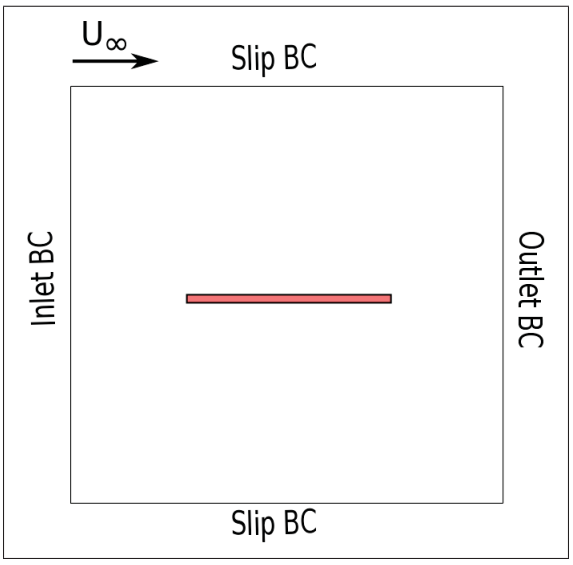

Figure 2.12 Illustration of the 2D discontinuity segment case.

[4]The source code plus an application case for OpenFOAM(2.2.1) can be found in the attached DVD-R and a tutorial is available in Appendix II.

In the case of the velocity discontinuity the proposed 2D validation case concerns a velocity discontinuity segment. The segment is immersed in a uniform flow as shown in Figure 2.12. A uniform velocity discontinuity is applied in the flow direction. The solution of the problem is a combination of the incoming flow plus the perturbation flow caused by the discontinuity segment. The exact analytical solution is given by the Biot-Savart law applied to a point P.

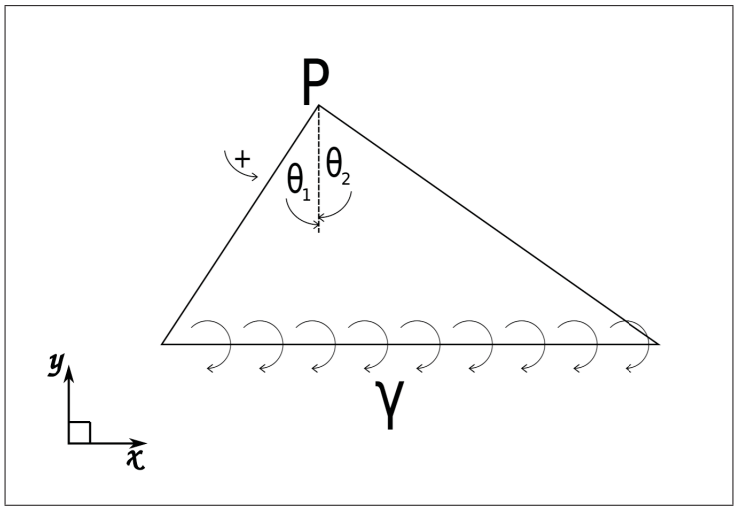

Figure 2.13 Biot-Savart law applied to one point P.

$$\vec{U}(\vec{r}) = \int_{-\infty}^{\infty} \frac{\Gamma}{4\pi} \frac{d\vec{l} \times \vec{r}}{|r|^3} \tag{2.24}$$

Figure 2.13 illustrates the analytical solution. In the same manner, Equation 2.24 shows the Biot-Savart law applied to a point P where r is the distance between any point in the segment to a point P and l is the segment length. The appropriate integration is applied to Equation 2.24. It must be taken into account that since we have a uniform distribution of discontinuity: $\gamma = \Delta u$. Once this is done it is easy to deduce that the components of velocity of the perturbation flow $\vec{U}$ are given by:

$$du_P = \frac{\Delta u}{2\pi}(\theta_2 - \theta_1) \tag{2.25}$$

$$dv_P = \frac{\Delta u}{2\pi}(\ln|\cos(\theta_1)| - \ln|\cos(\theta_2)|) \tag{2.26}$$

In these equations the angles θ_1 and θ_2 represent the angles between the point P and the leading and trailing edges of the segment. An illustration can be found in Figure 2.13. A similar form of derivation can be found in Leclerc and Masson (2004).

Table 2.4 Boundary conditions and schemes for 2D velocity discontinuity segment case.

	Walls	**Inlet**	**Outlet**	**Scheme**
Velocity	slip	fixedValue	zeroGradient	first & second order
Pressure	zeroGradient	zeroGradient	fixedValue	second order

When velocity discontinuity was studied, several mesh resolutions were used to analyze the case. The results presented in the following section correspond to a 700x700 mesh, where 100 cells where used to discretize the AS.

2.2.2.3 Results: 2D Velocity discontinuity segment with uniform distribution

The first thing that is noticed in the validation case is that in the case of the coarser meshes (that were first used), the streamlines around the segment of discontinuity were not perfectly circular. This is noticed especially around the trailing edge of the segment. This behaviour tends to diminish as the mesh resolution is increased. Nevertheless, for a fine mesh of 490,000 nodes, even though the streamlines are close to perfect, a small wake-like behaviour can still be observed. Figure 2.14 shows the perturbation flow both for the analytical and numerical solutions. This corresponds to a fine mesh, where the AS is represented by the black bold line.

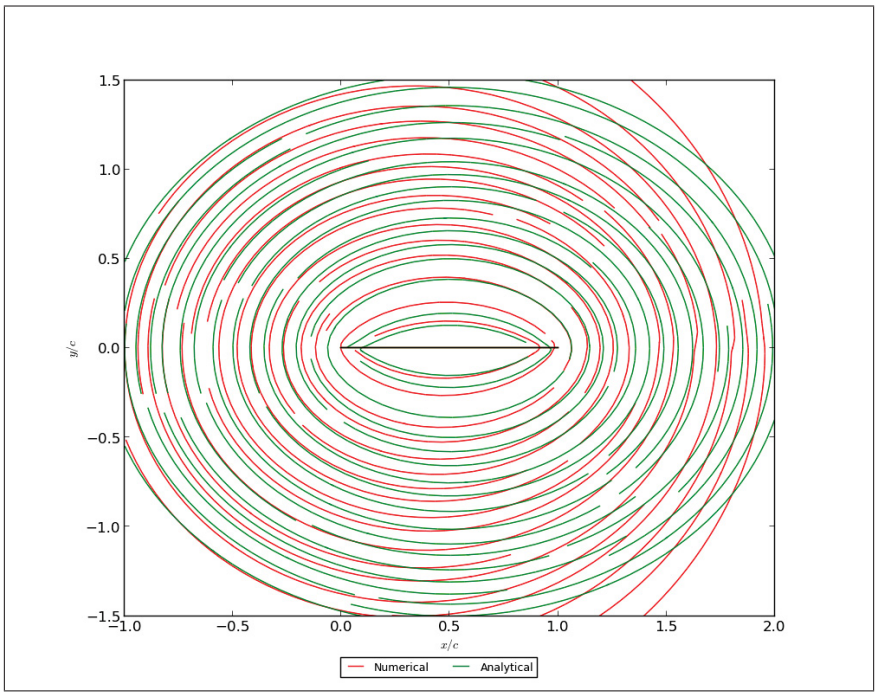

Figure 2.14 Streamlines of perturbation flow.

It has been determined that a much finer mesh resolution is needed to guarantee grid independence for the solution. This particularly applies when using a uniform distribution of discontinuity. Following the recommendation made by Sibuet Watters and Masson (2010), tests were conducted using a parabolic distribution. It has been verified that distributions respecting continuity such as this one do not require as fine a mesh resolution to obtain mesh independence. This is in contrast to the uniform distribution.

In order to evaluate the impact of the flow on the perturbation flow solution, simulations have been performed for different incoming flows. The angles used were: 0°, 10°, 30°, and 60° with respect of the AS. This can be seen in Figure 2.15. Several segment sizes and prescribed circulations were also used. The external force induced by the vorticity segment on the flow has been evaluated. It has been verified that the normal component is in excellent agreement with the Kutta-Joukowski relationship:

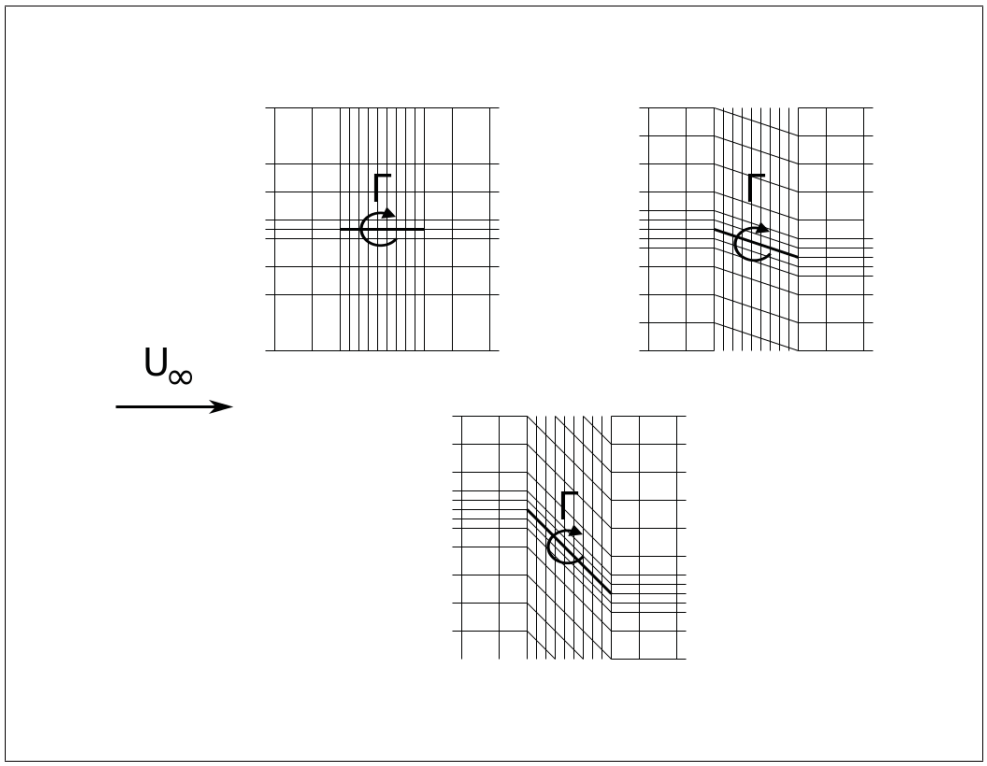

Figure 2.15 Different configurations for the discontinuity segment.

$$L = \rho U_\infty \Gamma \tag{2.27}$$

In fact, it was verified that the current implementation gives values within 5% of this relationship. The total circulation is given by the uniform distribution calculated with Equation 2.20. The tangent component of the force in this case is very close to zero.

Another limitation is encountered when implementing the velocity discontinuity. As seen in the mathematical model, the average value of velocity upstream and downstream of the AS is required in order to calculate the system of forces attached to the AS. The function allowing for extraction of the information of an arbitrary node in OpenFOAM is not parallelized. Consequently, the code will only run in one processor until this inconvenience is overcome.

CHAPTER 3

APPLICATION OF THE ACTUATOR SURFACE

The present chapter concerns the application of the AS method, as well as its performance in ideal and experimental cases. The first case to be analyzed is the case of the wing in translation. During this case the code's performance will be compared against Prandtl's lifting line theory. This is a necessary step before the final application. The final application case is the case of the wind turbine (or wing in rotation). The performance of the code is then compared with the experimental data. The data to be compared in this case are those of the TUDelft rotor.

3.1 Finite wing in translation

3.1.1 Mathematical Considerations

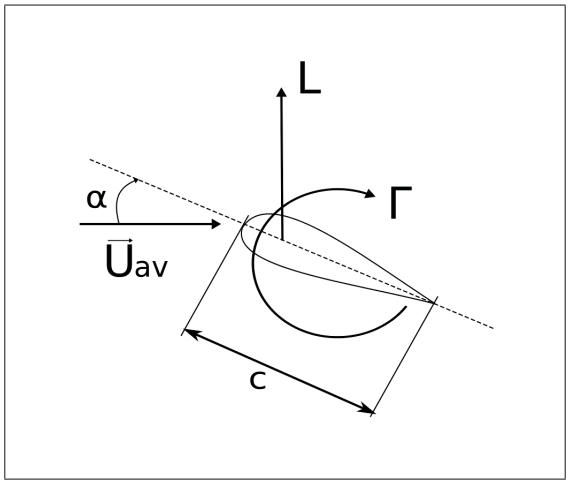

Figure 3.1 Wing cross section.

When talking about the wing in translation certain mathematical considerations must be taken into account. These mathematical concepts are based on thin airfoil theory, such that the AS method closely resembles vortex methods. In the present case of study nodes are located at mid-chord of the wing segments. The nodes will function as control points where the induced angle α_i is evaluated.

$$\alpha_i = \tan^{-1}(-w_{av}/U_\infty) \tag{3.1}$$

In order to calculate the loading of the AS, only the lift characteristics of the lifting device are needed. The lift values are directly related to the operating conditions of the wing. A relationship between the angle of attack, the pitch angle β, and the lift coefficient cl is given by:

$$cl = 2\pi(\beta - \alpha_i) \tag{3.2}$$

The circulation around the wing segment is then obtained. The circulation can be regarded as a function of the incoming flow U_∞, the lift coefficient and the chord length c:

$$\Gamma = \frac{clU_\infty c}{2} \tag{3.3}$$

3.1.2 Discontinuity distribution

A seen during the implementation of the velocity discontinuity, the simplest approach to distribute the discontinuities would be to use a uniform distribution. Nevertheless this type of distribution induces instabilities in the numerical solution. Once the circulation around the airfoil is calculated, another kind of distribution respecting continuity is implemented. The distribution that will be used during this case is a parabolic distribution:

$$\Delta u_P = \frac{6\Gamma}{c^3} x_P(c - x_P) \tag{3.4}$$

Where x_P is the distance between a point P and the leading edge of the wing segment. In the case of a finite wing in translation a second velocity discontinuity appears tangent to the AS (Δv). In other to evaluate the discontinuity the first condition of zero vorticity flux must be

respected. Applying the constraint given by Equation 2.4 the following expression is obtained:

$$\Delta v_P = \frac{6\Gamma}{c^4}\frac{\delta c}{\delta y}(x_P^3 - cx_P^2) + \frac{1}{c^3}\frac{\delta \Gamma}{\delta y}(3cx_P^2 - 2x_P^3) \tag{3.5}$$

In special cases when the taper ratio of the wing is equal to 1, the first term of the right side of the equation becomes null. This happens as the chord length remains constant along the wing span. Taking the new mathematical considerations and the parabolic distribution into account, a new version of the solver is implemented. The solver is named simpleFoamWT[1], which stands for simpleFoam Wing in Translation.

3.1.3 Validation Case

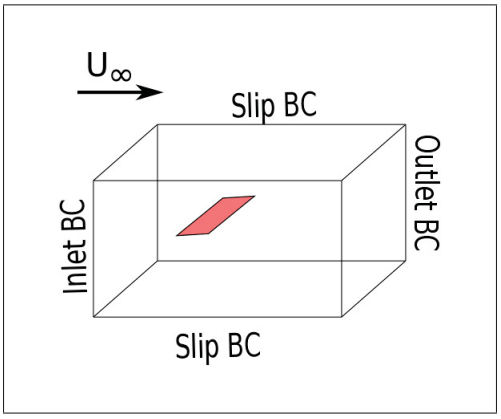

Figure 3.2 Illustration of the wing in translation.

Figure 3.2 illustrates the problem of the finite wing in uniform translation. In this case of study the finite wing is defined as a rectangular surface parallel to the incoming flow. The wing lies on the XY plane. The geometric characteristics of the wing include a constant span length b of 10m (for all cases). Only a taper ratio of 1 was studied, with a combination of 4 aspect ratios AR(4; 6; 8; 10). The AR is equal to the ratio of b^2 to the wing area.

[1]The source code plus an application case for OpenFOAM(2.2.1) can be found in the attached DVD-R and a tutorial is available in Appendix III.

In all simulations the wing is immersed in a uniform incoming flow of 50m/s. In some cases a flow of 10m/s was also studied. The domain is a cube whose side length is equal to 30b. The wing is located at 10b downstream of the inlet in the middle of the YZ plane.

The simulation results are then compared with Prandtl's lifting line theory. The exact semi-analytical solution can be found in the work of Anderson John (2005) and Panton (1996). Three grid resolutions were used for the simulations: a coarse mesh (Mesh 1), a medium mesh (Mesh 2) and a fine mesh (Mesh 3). Details on mesh characteristics as well as the number of nodes used to discretize the AS can be found in Table 3.1.

Table 3.1 Mesh details for the wing in translation.

	Mesh 1	**Mesh 2**	**Mesh 3**
Number of nodes for domain discretization	57x48x60	80x62x92	140x98x124
Number of nodes in the AS	5x20	13x32	17x40

The mesh is refined within proximity of the wing. Both first and second order convection schemes were used, as well as a similar set of boundary conditions used in previous cases.

Table 3.2 Boundary conditions and schemes for the wing in translation case.

	Walls	**Inlet**	**Outlet**	**Scheme**
Velocity	slip	fixedValue	zeroGradient	first & second order
Pressure	zeroGradient	zeroGradient	fixedValue	second order

The study of the wing in translation is a necessary step before the analysis of the wing in rotation. If the code performs correctly, it is assumed that a similar behaviour will occur for the study of a wind turbine.

3.1.4 Results: Wing in translation

The results presented in this section correspond to the case of a wing of AR 4 and an incoming velocity of 50m/s. Mesh 3 was used to obtain these results. Figure 3.3 presents a downstream view of the streamlines near the wings tip vortices. This figure was taken in a slightly inclined

plane for better appreciation of the streamlines. A 3D view as seen in the post-processing sofware paraView can be appreciated, where 2 downstream distances have been marked. The distances that can be seen correspond to 5b and 10b downstream of the AS, which is represented by a red surface. This is very interesting to observe, as it can be confirmed that the model is able to represent and trace wind behaviour caused by the development of tip vortices. Figure 3.4 portrays the influence of the convection schemes. The figure shows the evolution of the component w_{av} over the discretized wing. It is observed that the value varies from positive to negative in the direction of the flow.

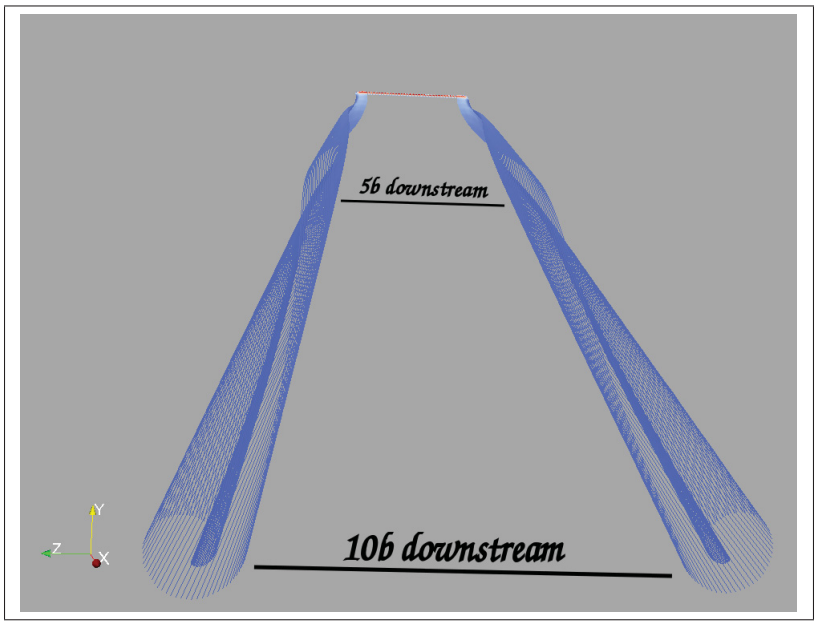

Figure 3.3 Incline view of streamlines near wing tip vortices (Incoming flow of 50m/s - Wing of AR 4).

Intense variations of w_{av} can be observed at the tip of the span. Both schemes seem to handle these variations adequately, so it is concluded that both convection schemes give an adequate representation of the flow. Nevertheless the second order convection scheme provides a much more accurate and stable solution to the problem. This can be observed in the flow in both the leading and trailing edges, where the flow values appear much clearer when solved by the second order scheme.

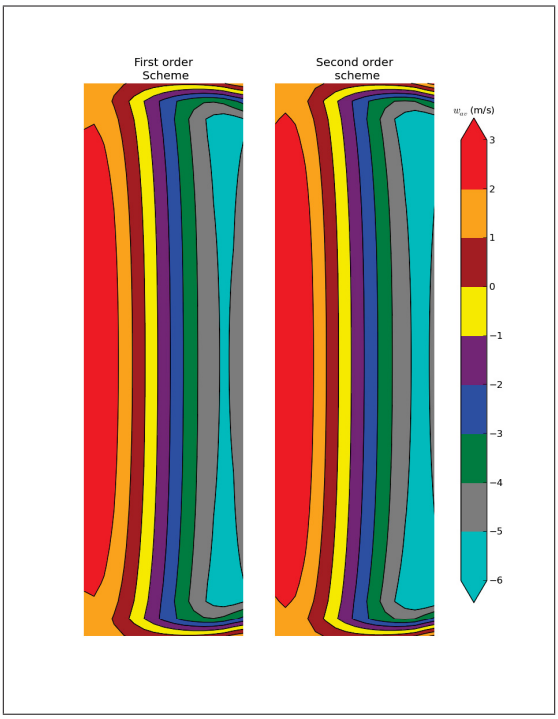

Figure 3.4 Convection scheme effect on w_{av} isocontours (Incoming flow of 50m/s - Wing of AR 4).

The induced flow angle is also subject to study. The induced flow angle is defined as the opposite of the induced angle α_i. Figure 3.5 shows the induced flow angle for Mesh 1, Mesh 2 and Mesh 3. The induced flow angle is shown along the span at mid-chord. Figure 3.5 presents the results for both first and second order convection schemes. The results are then compared with Prandtl's lifting line.

An excellent agreement is observed between the numerical and analytical results. The best results are given by the second order scheme for Mesh 2 and Mesh 3. It is therefore concluded that an intermediate mesh resolution is sufficient to obtain quality results. A new limitation appears at this point, as an odd number of cells along the chord are needed to discretize the wing. The odd number of nodes will ensure that the induced angle α_i is evaluated at mid-chord. This is required by the model. This is a limitation related to OpenFOAM, as the program will not allow nodes to be located at the cell faces.

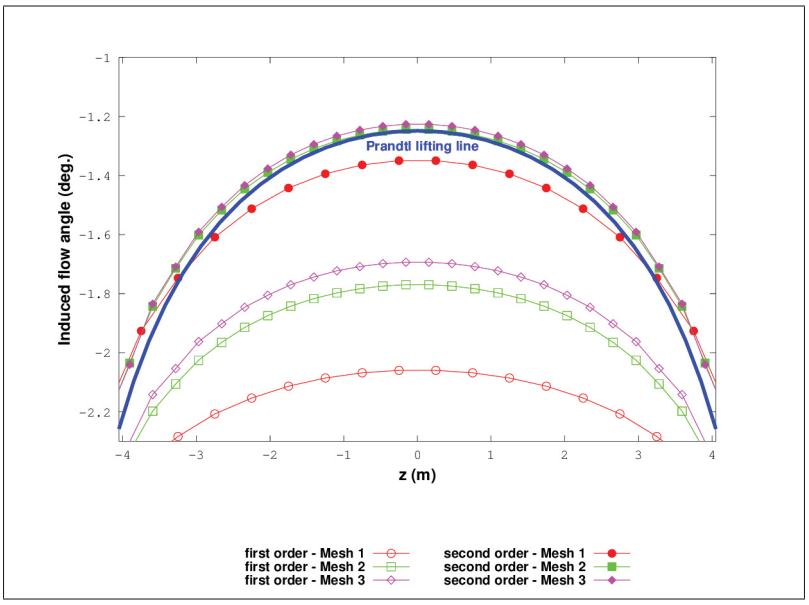

Figure 3.5 Spanwise distribution of induced flow angle compare with Prandtl's lifting line.

In some cases simulations were also performed for wings immersed in an incoming flow of 10 m/s. The induced flow angle remained the same for wings of equal AR. This confirms that the induced angle is a function of the geometry of the wing.

The code used in the case of the wing in translation is ready take real airfoil data. The code can also be applied in real cases for wings and other lifting devices. Nevertheless the main objective of this project is the application of the AS for wind turbines. Therefore no further simulations were performed for this particular case.

3.2 Wing in rotation (wind turbine)

3.2.1 Mathematical Considerations

In the case of the wind turbine, a rotating frame of reference method was chosen. When confronted with the rotating frame, certain terms must be taken into account. In this method the blades appear fixed. The apparition of centrifugal and Coriolis forces is observed as a

consequence of rotation. These forces must be added to the Navier-Stokes equations. Both centrifugal and Coriolis forces collapse into a single term in the inertial frame of reference:

$$\int_A \nabla \cdot (\vec{U}_{rel} \otimes \vec{U}_I) dA + \int_A \vec{\Omega} \times \vec{U}_I = -\int_{V_{up}} \nabla p dV - \int_{V_{dn}} \nabla p dV + \int_{A_{AS}} f_t dA \tag{3.6}$$

When observing Equation 3.6 different velocities are noted: the velocity in the rotating frame $\vec{U}_{rel}$, the velocity in the inertial frame $\vec{U}_I$ and the rotational velocity $\vec{\Omega}$, where:

$$\vec{U}_{rel} = \vec{U}_I - \vec{\Omega} \times \vec{r} \tag{3.7}$$

In Equation 3.7, r represents the radial distance from centre of rotation to a point P. Since the case of study is no longer a wing in translation induced angle α_i is replced by the angle of attack ξ_i. The way the angle of attack is calculated must be reexamined. In the case of the wind turbine the incoming flow experienced by the airfoil corresponds to that of the relative frame. Blade element analysis is used, and the geometrical and operational characteristics of the turbine can be found:

$$\Theta_i = \tan^{-1}(-w_{rel,av}/u_{rel,av}) \tag{3.8}$$

$$\xi_i = (\Theta_i - \beta_i) \tag{3.9}$$

Θ_i being the blade segment's angle relative to the wind. β_i is the local pitch angle. With the help of experimental data, the lift coefficient (cl) that corresponds to the selected airfoil can be determined. The Kutta-Joukowski relation will serve to calculate the circulation around the airfoil.

$$\Gamma = \frac{cl \|U_{rel}\| c}{2} \tag{3.10}$$

The parabolic distribution of the velocity discontinuities presented in the previous section is conserved. Nevertheless, the way the forces are calculated must take into account the relative velocity:

$$\Delta p = \rho u_{rel,av} \Delta u + \rho v_{rel,av} \Delta v \tag{3.11}$$

In the case of the tangent forces a similar approach is used. The velocity is averaged as shown in Equation 2.7, but the velocity will correspond to that of the relative frame.

3.2.2 Validation Case

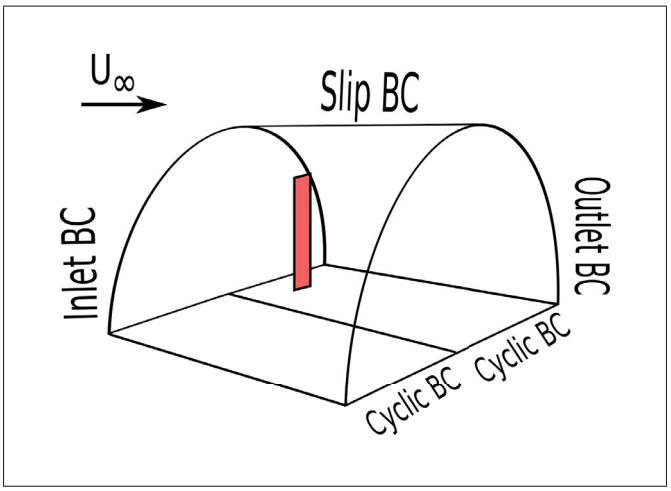

Figure 3.6 Illustration of the wing turbine case.

Lastly, the case of the wind turbine or wing in rotation is presented. For this case, a rotor designed by the Delft Technical University (TUDelft) is studied. The rotor characteristics can be found in Table 3.3.

The experimental data for the TUDelft rotor was collected using hot film anemometry. Further details about the experimental procedures and results can be found in the work of Haans *et al.* (2008) and Sant (2007). Figure 3.7 presents the mesh details for the rotor case, that was

Table 3.3 TUDelft rotor characteristics.

Number of blades	2
Airfoil section	NACA0012
Rotor radius R_T (m)	0.6
Blade root radius R_r (m)	0.18
Chord c (m)	0.08
Blade pitch angle β_i (°)	$\beta_i(r/R) = (6 + \beta_{i_{tip}}) - 6.67 \cdot r/R \quad 0.3 <= r/R <= 0.9$ $\beta_i(r/R) = \beta_{i_{tip}} \quad 0.9 < r/R <= 1$

constructed with the default application blockMesh. In accordance with the previous cases the grid is refined around the AS.

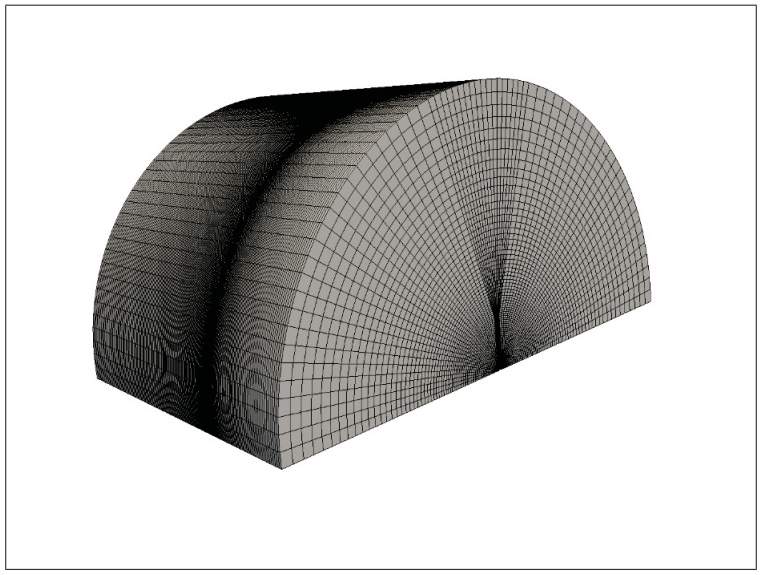

Figure 3.7 Mesh detail of the wing turbine case.

The domain for this particular case is a close reproduction of that found in the work of Sibuet Watters and Masson (2010). The domain is composed by a half a cylinder with a length of 3.5D, an inner radius of 1.5 10^{-2}D and an outer radius of 5D. Only one blade is modelled. A mesh of 200x65x54 nodes was used in the axial, azimuthal and radial directions, respectively. 17x20 nodes were used to discretize the AS. Since OpenFOAM only works with a Cartesian coordinate system, the inner radius is necessary to avoid a singularity

Table 3.4 Boundary conditions for the wind turbine case.

	Walls	**Inner radius**	**Inlet**	**Outlet**	**Cut**
Velocity	slip	slip	fixedValue	zeroGradient	cyclic
Pressure	zeroGradient	zeroGradient	zeroGradient	fixedValue	cyclic

Boundary conditions can be found in Table 3.4. Pitch angles of 0, 2 and 5 degrees were studied. A second order convection scheme was used, as well as a second order central differencing scheme for pressure. Experimental data of the airfoil lift coefficients corresponding to a Reynolds number of 150000, as seen in the work of Claessens (2006), was used. Only the lift characteristics are required to calculate the AS loading as seen in previous sections. An artificial viscosity was employed to ensure the stability of the simulation. It was verified that the value used did not influence the near wake results reproducing an inviscid flow. A range of 1-10 times the air viscosity was studied. Following a sensitivity analysis, it was decided that a value of 5 times the air viscosity was sufficient to guarantee the stability of the simulation. Further viscosity value analysis will be needed when simulating other turbines.

On a first attempt to simulate the turbine the solver SRFSImpleFoam was used. This particular solver utilizes a single frame of reference. The problem is solved in the rotating reference frame. It became evident that the simulation was unstable, as well as having very limited configuration options. Therefore this solver was discarded as a viable option. OpenFOAM also has a multiple reference frame (MRF) option incorporated in solver simpleFoam. In this case the problem is solved in the inertial frame. This approach was deemed more appropriate.

Since the MRF option is usually used for more than one rotating frame, some issues arise when simulating a single rotating frame. If the whole domain is selected as the rotating frame, the boundary conditions that are set in the inertial frame are ignored by the solver. Only the rotating forces are taken into account over the flow solution.

To overcome this inconvenience a layer of at least one cell must be left between the rotating frame and the boundary conditions. This configuration has been thoroughly studied in domains with and without obstacles. Different velocities were used when running simulations without obstacles. In all cases the axial velocity was equal to U_∞. The radial and tangential components

of velocity quickly converged to zero. Only in the presence of an obtacle, the effects of rotation were observed. It has been determined that this configuration does not affect the overall results of the simulation. The boundary conditions are at a sufficient distance where the effects of the rotor are null and the velocity is equal to U_∞. The resulting solver is named simpleFoamBR[2] (simpleFoam Blade in Rotation).

3.2.3 Results: Wind turbine

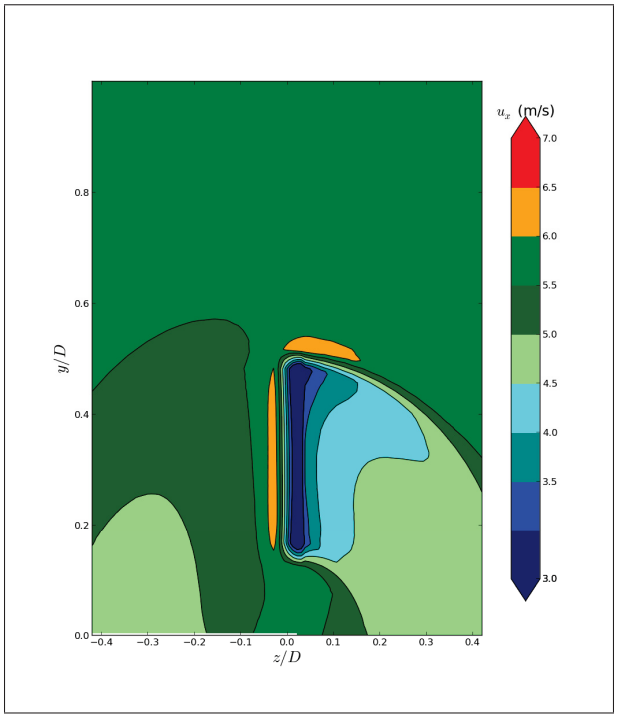

Figure 3.8 Axial component of Velocity.

The results of the simulation are compared against the experimental data from the TUDelft rotor. The results presented belong to a pitch angle β of 2 degrees and λ of 8. This value of λ corresponds to an incoming flow velocity of 5.5 m/s and a rotational speed of 700 rpm. Figure 3.8 shows the axial components of velocity, while Figure 3.9 shows the radial components (both in the inertial frame of reference). The axial components are taken in the plane of the

[2]The source code plus an application case for OpenFOAM(2.2.1) can be found in the attached DVD-R and a tutorial is available in Appendix IV.

rotor swept area. The radial components are taken in the axial plane at $\frac{\pi}{2}$, dividing the blade into two equal halves. The location of the blade in Figure 3.9 is represented by a bold black line.

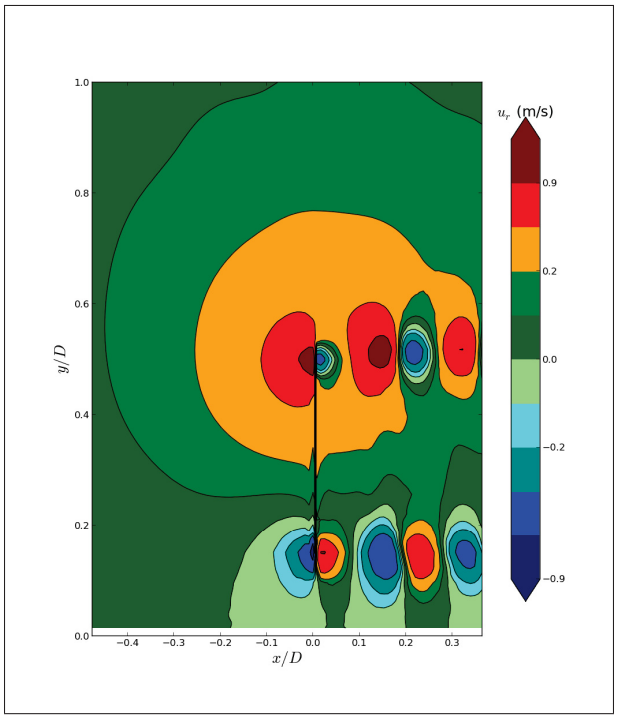

Figure 3.9 Radial component of Velocity.

Since the simulation has been performed in a steady state solver, both figures represent the flow when the blade reaches the vertical position. Only the non-rotating components of velocity are shown. It has been confirmed that the model behaves in a similar fashion as that implemented by Sibuet Watters and Masson (2010). It can therefore be affirmed that the model correctly represents the following parameters:

- Effects of the AS on the flow upstream from the rotor.
- Velocity discontinuities.
- Emanating vortical structures from the tip and root areas of the rotor.
- Effects of rotation.

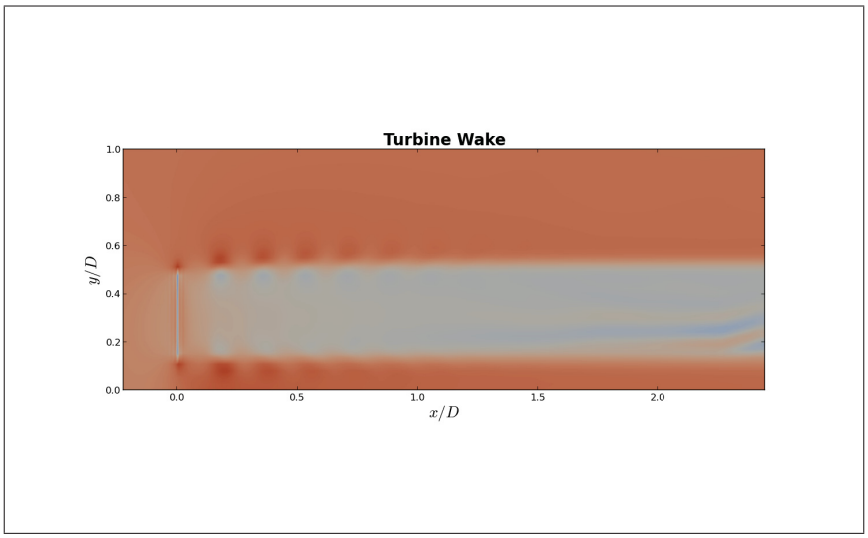

Figure 3.10 Turbine wake.

Figure 3.10 shows the turbine wake and its development. In the immediate vicinity of the blade the tip vortices can be observed. Tip vortex brake-down can be viewed further downstream until reaching the far wake. In the far wake, the wake is fully developed. It is also quite interesting to observe that in Figure 3.9, the tip vortices' trajectories are portrayed. This allows the AS model to trace such structures.

Figure 3.11 shows the radial distribution of the axial component of velocity at the centre of the blade. The experimental measurements are compared with the numerical results taken at $3.5x10^{-3}m$ upstream from the AS. The area in close proximity to the AS is an area of strong velocity variations. After a sensibility analysis it was determined that the distance at which the numerical data were sampled is the most appropriate and is remarkably close to the experimental measurements, especially at mid-span.

Lastly, a comparison between the model and measurements of the near wake region are shown in Figures 3.12, 3.13 and 3.14. The values are taken at planes parallel to the rotor swept area. These values are then plotted with respect to their azimuthal position θ. Axial, tangential and radial components of velocity are compared at 0.03D, 0.05D and 0.075D downstream of the rotor. All three figures show three radial locations: r/R = 0.4 (root area), 0.7 (mid-span area), and 0.9 (blade tip area).

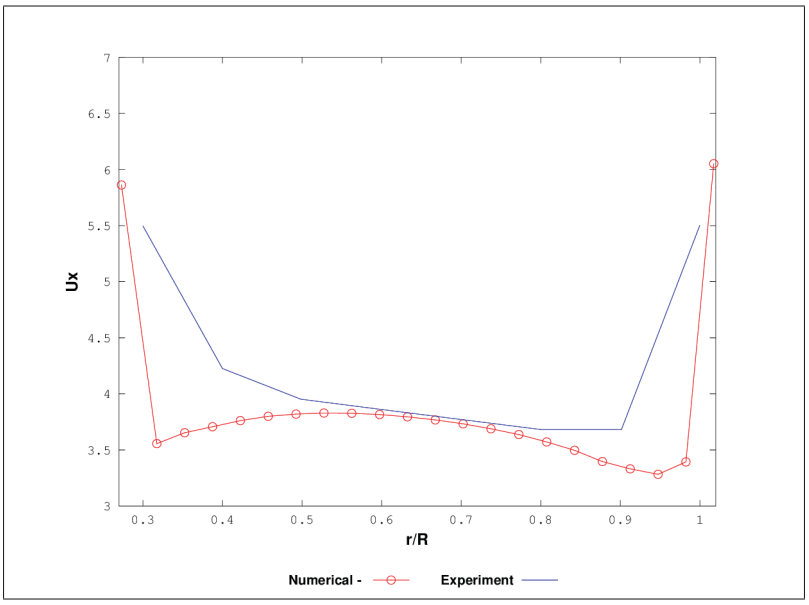

Figure 3.11 Radial distribution of axial velocity at the middle of the blade

Differences can be observed between the simulation and experimental results. There are several reasons for the disagreements between the model and the experimental data, namely:

- The effect of rotation in blade aerodynamics.
- The absence of solid boundary for the blades.
- The hub and tower are not modelled during the simulation.
- The blockage effect is not taken into account by the model.
- The differences between the real turbine and the AS modelled in OpeFoam. The AS is a porous surface, while the turbine is not completely porous.
- Finite volume method used by OpenFOAM.
- Uncertainties related to the simulation, domain size, mesh, number of cells used to discretize the blade, convection scheme, boundary conditions etc.

Nevertheless it is remarkable how the shape of the azimuthal distribution of velocity for both the simulation and the experiment behave in a similar manner. This similar behaviour can

be observed in the distance from root to tip. The same can be concluded with respect to the downstream distance.

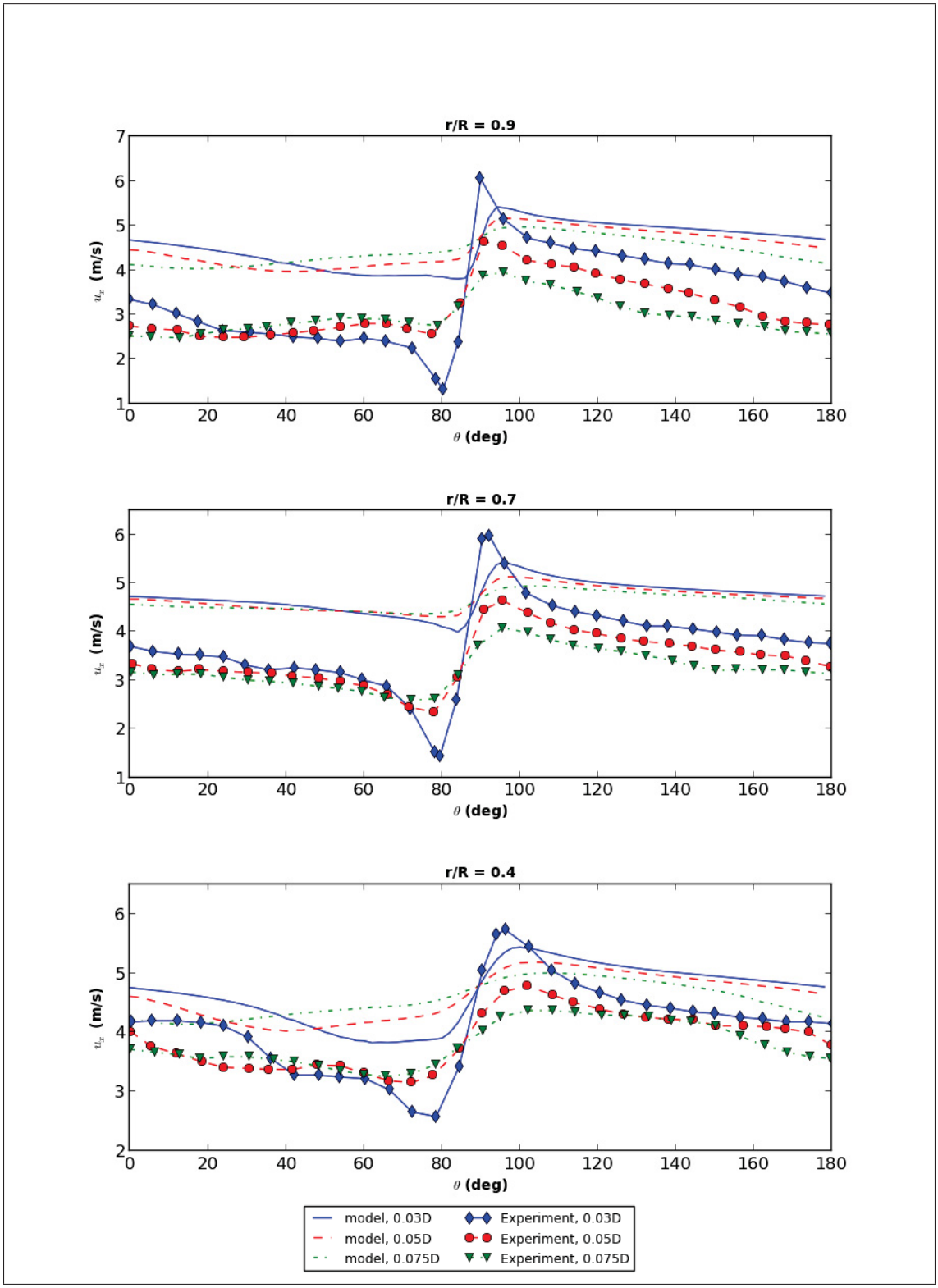

Figure 3.12 Axial component of Velocity as a function of azimuth for different radial locations.

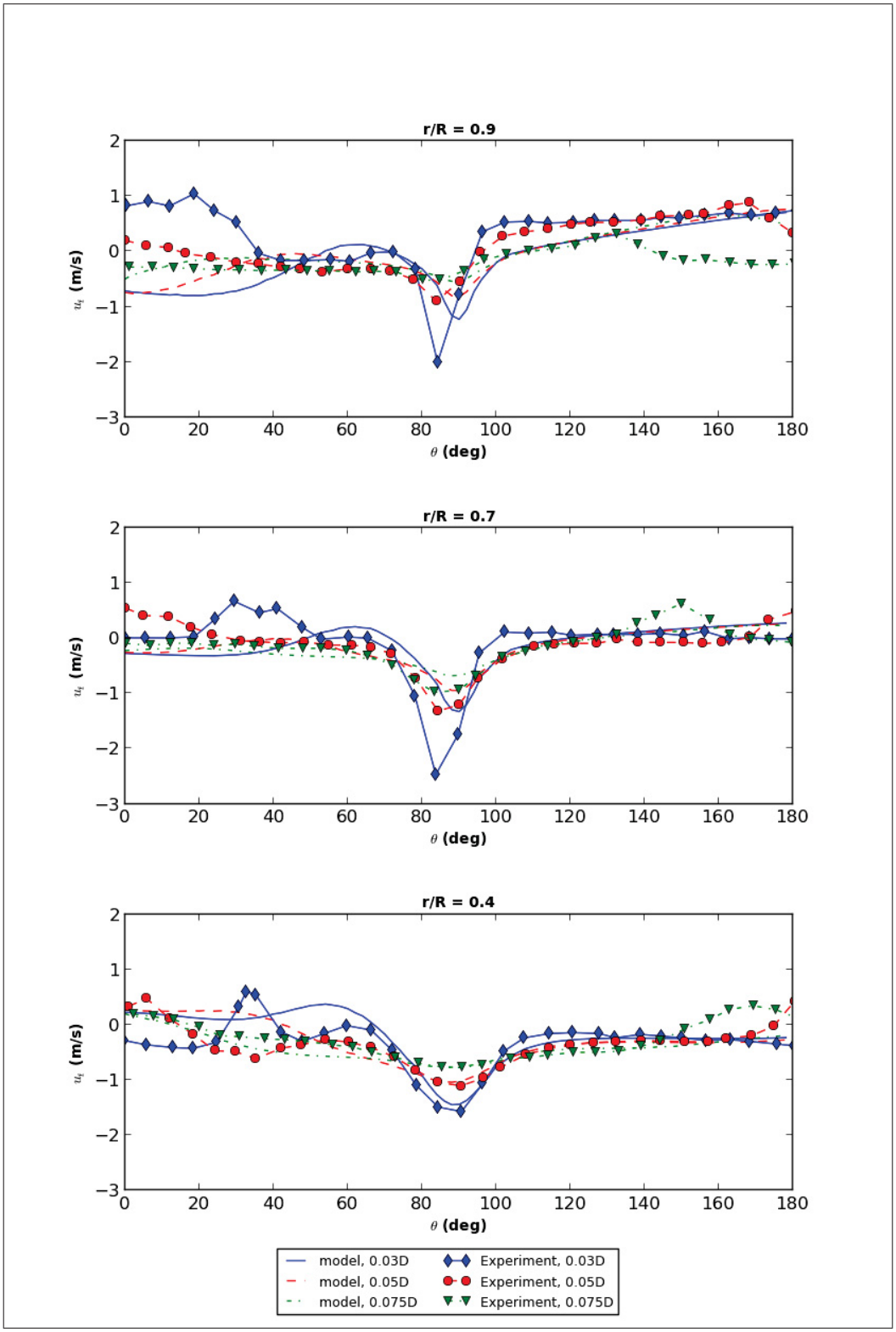

Figure 3.13 Tangential component of Velocity as a function of azimuth for different radial locations.

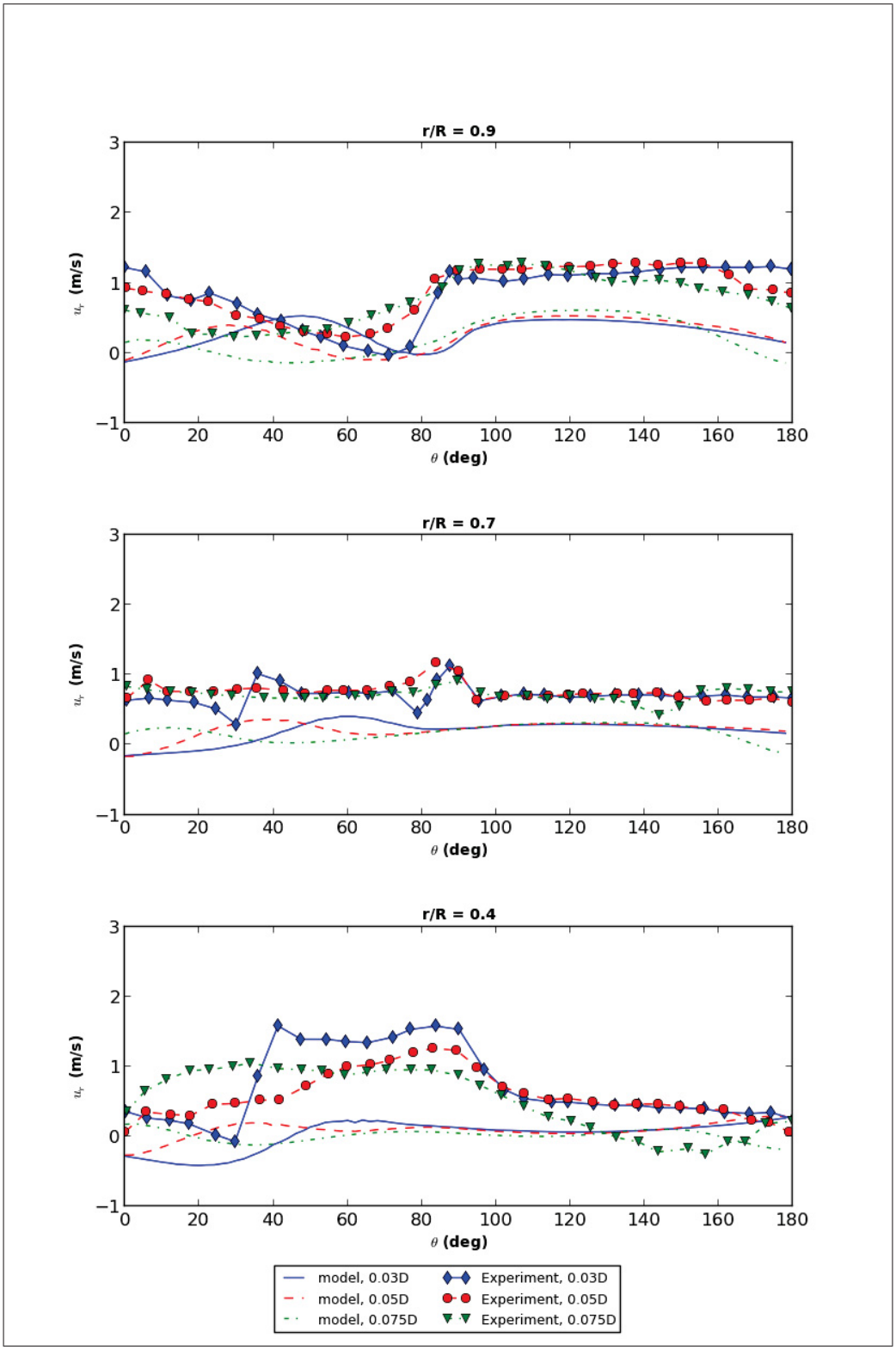

Figure 3.14 Radial component of Velocity as a function of azimuth for different radial locations.

CONCLUSION

The main objective of the investigation was to implement and validate the Actuator Surface model in the open source program OpenFOAM. This objective has been attained. The AS is defined as a porous surface that carries velocity discontinuities tangent to the surface. A system of forces is attached to the surface, resulting in a pressure discontinuity normal to the surface. The velocity field remains continuous in the direction of the pressure jump. The AS can also be described as an infinitely thin vortex sheet producing lift forces. The forces are related to the circulation characteristics of said vortex sheet. The implementation of the AS technique resulted in a model able to represent lifting devices. These devices can be in translation but most importantly in rotation. The modelling of wings and rotors has been achieved. Inviscid flow characteristics have been reproduced around the blades of a wind turbine. Therefore the study wind turbine wake aerodynamics was possible.

The investigation has been divided into two main sections. Firstly, Chapter 2 of the thesis, which concerned the implementation and validation of the AS. The mathematical model of the Actuator Surface developed by Sibuet Watters and Masson (2010) (which can also be found in the original thesis of Sibuet Watters (2009)), was successfully implemented in OpenFOAM. OpenFOAM is a free, open source CFD package, whose flexibility and capabilities make it one of the most powerful CFD tool kits available. The implementation of the AS model occurred in two stages:

- First the pressure discontinuity was implemented. To do so explicit correction terms were inserted in the steady Navier-Stokes equations in their inviscid form. Afterwards, the pressure discontinuity was then validated against two cases. In both cases an analytical solution was available. The infinite plane case and the well-known actuator disk case were in excellent agreement. Similar results were observed between the numerical and analytical solutions.
- The second stage of the implementation was the velocity discontinuity. In this case, tangent forces are also included in the model. The mass fluxes must account for the velocity

discontinuity. The implementation of the velocity discontinuity was then validated. This was done against the case of a 2D segment carrying a uniform velocity discontinuity. The analytical solution is given by the Biot-Savart law. Once the analytical solution was obtained, it was compared with the numerical perturbation flow results. It was observed that a fine mesh yields better results. The need to use distributions respecting continuity, such as the parabolic distribution, was confirmed.

The second section of the thesis is presented in Chapter 3. This chapter is dedicated to the application of the AS. The main objective is to apply the AS to the model a wind turbine. Before doing so, however, the model was tested by applying to the case of a finite wing in translation. This chapter is also divided into two main sections, as described below:

- The first application of the AS is to model a wing in uniform translation. This includes implementation of a parabolic distribution. This distribution is used to calculate the velocity discontinuities and the system of forces attached to the AS. A grid sensibility analysis was conducted to determine the appropriate mesh resolution to discretize the AS. The selected mesh minimizes the mesh dependency of the results. The results were then compared with Prandtl's lifting line theory. Remarkable agreement was found using medium and fine mesh resolutions. Results regarding first and second order convection schemes were also compared. It was determined that a second order convection scheme more accurately represented flow characteristics. The properties studied included the induced velocity, as well as the induced angle of attack α_i.
- The second part of Chapter 3 is the application of the AS to model a wind turbine. The model was applied to the experimental rotor of the Technical University of Delft (TUDelft). A rotating frame of reference approach was used. The flow was solved in the inertial frame of reference. Centrifugal and Coriolisis forces collapse in one single term in the inertial frame. Some changes were made in order to calculate the angle of attack by applying wind turbine theory found in the work of Manwell *et al.* (2010). Experimental lift data were used from the airfoil NACA0012 for a Reynolds number of 150000. The results were then compared with near wake measurements. It was observed

that the model reproduces tip vortices. Similar behaviour of the velocity components of the near wake was also observed.

The results observed when simulating the TUDelft rotor were very promising, as the model produced comparable results to other simulations methods.

Main Contributions

- The Actuator Surface model based on the mathematical model found in the work of Sibuet Watters and Masson (2010) was implemented. This was achieved by inserting explicit correction terms in the Navier-Stokes equations.
- The AS model was applied for 2D and 3D cases. A finite volume method in the open source program OpenFOAM was used. The method respects the conservation principles of momentum, energy and vorticity.
- The AS model was validated and applied in the form of analytical and experimental cases. This led to implementation of the following applications:
  - **SimpleFoamPJ**: Application used to model an infinite plane & actuator disk carrying a uniform pressure discontinuity.
  - **SimpleFoamVD**: This application is used to model a 2D segment of uniform velocity discontinuity, also known as the infinite wing.
  - **SimpleFoamWT**: In this application a finite wing in uniform translation is modelled. A parabolic distribution was used to calculate the velocity discontinuities.
  - **SimpleFoamBR**: Using a rotating frame of reference, this application solves the problem of a wing turbine. This is done by taking experimental lift data in order to calculate the system of forces. The forces are related to the total circulation of the segments.
- The implemented model is able to reproduce flow characteristics such as the induced velocity and the induced angle of attack. These results are in excellent agreement with Prandtl's lifting line theory.

- The implemented model is able to reproduce wind turbine vortical structures and tip vortices. These structures are in good agreement with the experimental wake measurements. In this work the experimental rotor of the Technical University of Delft was used as a reference case. The results were highly encouraging.

RECOMMENDATIONS

The implementation of the Actuator Surface technique in OpenFOAM is a great accomplishment. The various applications and their results show that this is a promising technique for the modelling and study of wind turbine near and far wake aerodynamics. Nevertheless, these accomplishments are not without their challenges and limitations. The improvement and future development of the applications mentioned in this thesis are a natural next step.

The current implementation of the AS technique requires further modification and improvement. As seen in the thesis, the pressure gradient correction terms are conceived for a uniform grid. Correction terms taking into account a non-uniform mesh would make the application much more flexible and simplify the mesh handling. Furthermore, correction terms for different discretization schemes other than a second order central differencing could be implemented.

The parallelization of the model is also a crucial step to be able to take advantage of the calculation power of OpenFOAM. Online forums and articles discuss the subject, including the function that allows the extraction of information of a random node. Nevertheless, this should be studied carefully as other problems may arise during the parallelization process.

Another development would be the inclusion of turbulence models and transient terms in the method. This is a delicate subject as different modifications and correction terms may have to be made for the transient term and for each turbulence model. This is necessary to account for the velocity discontinuities. Special attention should be paid to the viscous effects of the flow near the blades, as they may affect the creation of tip vortices.

Further comparison with classic models such as vortex methods, actuator disk and other volume force-based methods as well as experimental turbines is needed. The modelling of the tower and hub should be included to improve the quality of the results. The inclusion of drag characteristics of the airfoil during the calculation of the turbine loading should be considered.

OpenFOAM only works in Cartesian coordinates, although if a cylindrical coordinate system could be implemented, the need for domain alterations such as the inner radius could be discarded.

Lasty, the AS seems to correctly model the near and far wake characteristics of the turbine. This could be a step towards the implementation of a unified model for near and far wake predictions. Other interesting applications could be implemented such as the modelling of wind farms using the AS model.

APPENDIX I

SIMPLEFOAMPJ TUTORIAL

The following is a tutorial on how to use the application simpleFoamPJ. In this application, only the pressure discontinuity of the AS method is implemented for three-dimensional cases.

1. Solver Inputs

First a case directory must be created. Figure-A I-1 shows an example of the case directory for the Infinite plane case.

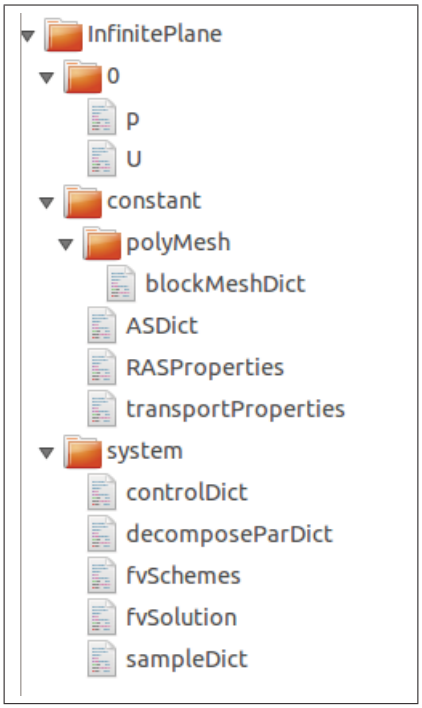

Figure-A I-1 Case directory.

This is a standard OpenFOAM case directory with the necessary input files for the application to function correctly. Next, the most important input files will be reviewed.

1.1 Mesh handling

First, the case must be set up, whether it be an infinite plane or an actuator disk. In the poly-Mesh directory, the file blockMeshDict is found. Follow the instructions in the Open- FOAM user guide to generate the case mesh.

In the case of an infinite plane the domain is a rectangular channel, as seen in Figure-A I-2. Following the method described in the body of the thesis four rows of cells must be uniform and cover the entire plane where the AS located.

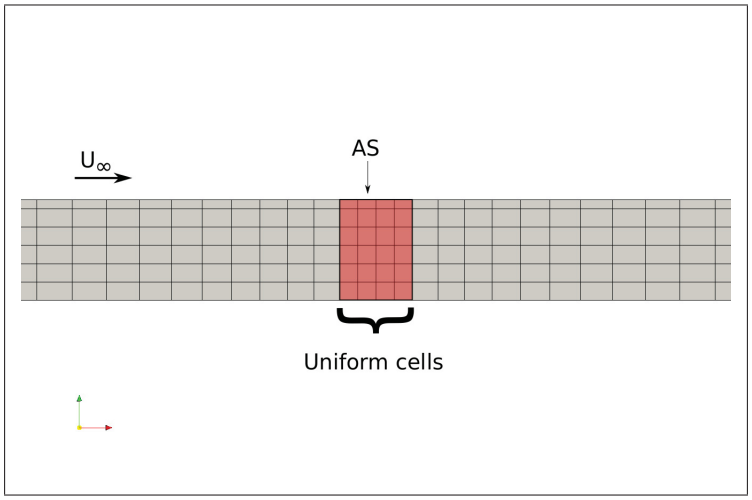

Figure-A I-2 Rectangular channel mesh example for infinite plane case.

In the case of the actuator disk the domain is a cylinder, and only four rows of cells inside the diameter of the disk must be uniform. Figure-A I-3 shows the actuator disk mesh in red. The rest of the domain can be handled at will. Both examples are exaggerated illustrations of the grid. As a recommendation in the case of the actuator disk, the greater the distance between the AS and the boundaries, the better the results will be obtained.

1.2 Boundary Conditions

Once the grid is generated the boundary conditions must be assigned. To do this go to the time 0 directory. There the file U and p are found. In both cases (infinite plane and actuator disk) the boundary conditions presented in Table-A I-1 were used (the boundary conditions are presented in OpenFOAM notation. Chapter 2 of the thesis provides an explanation as to the effect of each one).

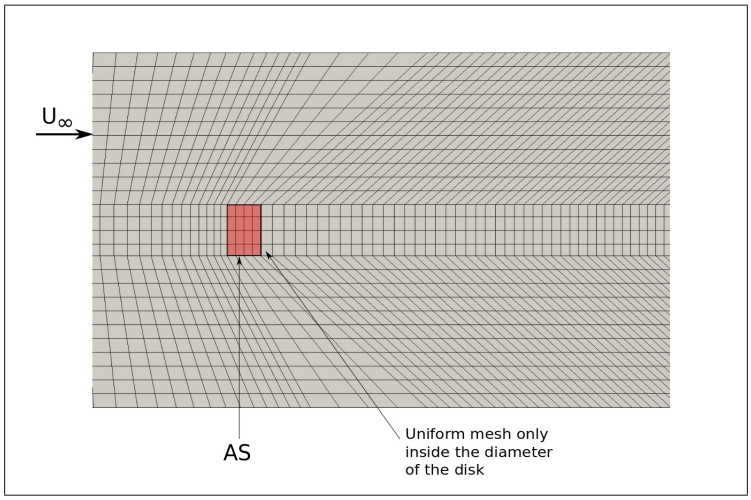

Figure-A I-3 Deformed mesh example for the actuator disk case.

Table-A I-1 Boundary conditions

	Walls	**Inlet**	**Outlet**
Velocity	slip	fixedValue	zeroGradient
Pressure	zeroGradient	zeroGradient	fixedValue

1.3 Interpolation schemes

With regard to the interpolation schemes, the file fvSchemes is located in the system directory inside the case file. Special attention must be paid to the pressure and velocity schemes, which must be set as presented in Figure-A I-4. As seen in the thesis the correction terms are only

conceived for a second order central differencing scheme. In OpenFOAM notation this is the linear scheme. If the pressure and velocity schemes are not properly set , the results will not be accurate.

```
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
//    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss upwind;
}

laplacianSchemes
{
    default         none;
    laplacian((1|A(U)),p) Gauss linear corrected;
}
```

Figure-A I-4 fvSchemes.

As seen in Figure-A I-4, for the velocity three different schemes can be used, namely a first order scheme **upwind**, and two second order schemes: **linearUpwind** (a mix between linear and upwind) and **SFCDV** (self-filtered central differencing for vector fields). If the linearUpwind scheme were to be used, simply replace **upwind** by **linearUpwind grad(U)** and uncomment the grad(U) interpolation scheme.

1.4 New Dictionary

A set of new variables is needed to run the application; a new dictionary must be created. This dictionary is called ASDict in this case and is located inside the constant directory. The easiest way to create ASDict is to copy another dictionary and change the name. In this dictionary certain variables must be defined, as shown in Figure-A I-5.

In the case of the actuator disk, $xmin$ and $xmax$ determine the location of the AS cell in the direction of the pressure jump. yo and zo set the location of the centre of the actuator disk, and together with the $tipRadius$ variable select the nodes inside the AS. If desired the $hubRadius$ variable can be used to leave an unselected area in the centre of the disk. Use this only if the hub is to be modelled differently. If not simply set it to zero. In order to calculate the normal

Actuator disk

```
// * * * * * * * * * * * * * * * *
xmin            2;
xmax            2.1;
zo              0;
yo              0;
tipRadius       0.2;
hubRadius       0;

ct              0.01;
rho             1.225;
Vo              10;
// ******************************
```

Inifinite plane

```
// * * * * * * * * * * * * * * * * *
xmin            99.5;
xmax            100;
ct              0.01;
rho             1.225;
Vo              10;
/*DUMMY VARIABLES*/
zo              1;
yo              1;
tipRadius       1;
hubRadius       1;
// *********************************
```

Figure-A I-5 ASDict.

force the magnitude of the incoming flow Vo, the thrust coefficient ct and the density rho are required.

In the case of the infinite plane the AS covers the entire plane. Nevertheless the application requires the same variables where yo, zo, $tipRadius$ and $hubRadius$ become dummy variables, which will not have an effect on the solution. Note that since this solver works with incompressible flows the force is divided by the density to render its effect null. Note that the code is ready to be used with compressible flows. If desired open the file Ucorrection.H in the application directory and modify accordingly.

1.5 Other dictionaries

As seen in the case directory, other dictionaries are required as input for the application. An in-depth explanation on how to configure these files can be found in the user guide, but a brief explanation is presented:

- **RASProperties**: Located in the constant directory, since the turbulent term of the Navier-Stokes equations is not included, this dictionary has no effect but is required by OpenFOAM. For good measure the RASModel option is set to laminar and the turbulence to off.

- **RASProperties**: Located in the constant directory. Once again, this dictionary has no effect in the solution but is required. For good measure the viscosity is set to a very low value.
- **controlDict**: Located in the system directory. The number of iterations must be set to ensure a good convergence.
- **fvSolution**: Located in the system directory. Leave the default solvers, a convergence criterion can be set if necessary, as well as relaxation factors.
- **decomposeParDict**: Located in the system directory. To be used if the code is set to run in parallel.
- **sampleDict**: Located in the system directory. Used to take sample values for post-processing.

```
        if ((cellCenter[0] < xmax) &&
        (cellCenter[0] > xmin))
//      && (mag(r) < tipRadius) &&
//      (mag(r) > hubRadius))
        {
                ++numInside;

                AScells.append(cellI);
        }



        if ((cellCenter[0] < (xmax+xvar)) &&
        (cellCenter[0] > (xmin+xvar)))
//       && (mag(r) < tipRadius) &&
//      (mag(r) > hubRadius))
        {
                AS2ndrowcells.append(cellI);
        }
```

Figure-A I-6 Lines to comment for the infinite plane case.

2. Application particularities

SimpleFoamPJ is designed to be used for both the infinite plane and actuator disk cases. In the file createVariables.H, a series of lists have been created to save computational time. These

lists save the location of the cells where the discontinuities are going to be created. In the case of the infinite plane, comment these lines and recompile as shown in Figure-A I-6.

In the case of the actuator disk, leave untouched. Since the main objective of this application was to set up validation cases, no further development was performed. However, the application is ready to take all types of distributions such as the Goldstein distribution for propellers that can be found in the work of Goldstein (1929). Svenning (2010) implemented an actuator disk in OpenFOAM with this distribution. As seen in the literature review, Meyers and Meneveau (2010), Calaf *et al.* (2010) and Prospathopoulos *et al.* (2009) also have different distributions for actuator disks. To run the application simply go to the case file and type simpleFoamPJ in the terminal, or it may also be run in parallel.

APPENDIX II

SIMPLEFOAMVD TUTORIAL

The following is a tutorial on how to use the application simpleFoamVD. This application was designed for the two-dimensional velocity discontinuity validation case (also known as the infinite wing).

1. Solver Inputs

First a case directory must be created. Figure-A II-1 shows an example of the case directory for the 2D velocity discontinuity segment.

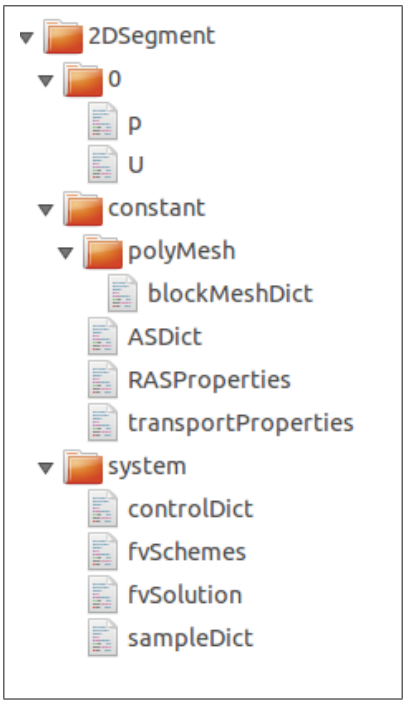

Figure-A II-1 Case directory.

This is a standard OpenFOAM case directory, with the necessary input files for the application to function correctly. Next, the most important input files will be reviewed.

1.1 Mesh handling

In the polyMesh directory the file blockMeshDict is found. Follow the instructions in the OpenFOAM user guide to generate the case mesh. OpenFOAM only works with three-dimensional meshes. In order to simulate a 2D case, for example in XY, only one dummy cell will exist in the Z direction. In blockMeshDict, all the patches belonging to the Z direction boundaries must be set as empty. In the case of the velocity discontinuity the AS is parallel to the flow. Figure-A II-2 shows an example of the mesh. As usual, four rows must be uniform normal to the surface.

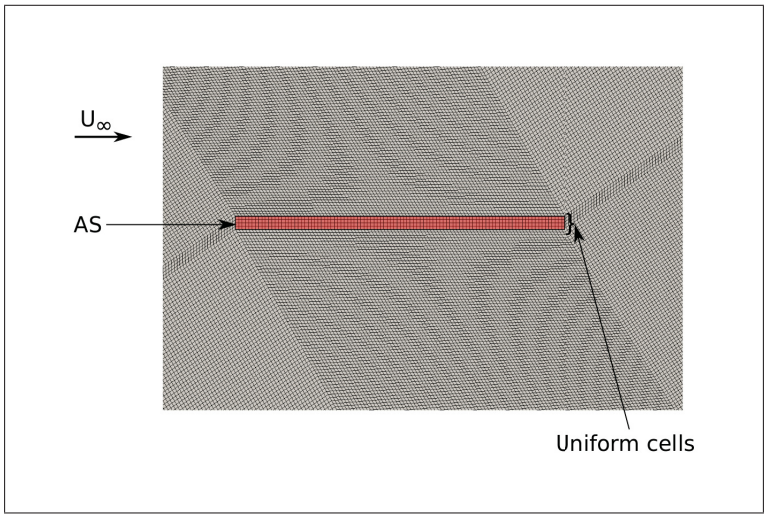

Figure-A II-2 2D segment mesh example.

If the orientation of the segment has to be changed, there are two options. Either the orientation of the segment is changed or the segment is left untouched and the orientation of the domain is changed. Even one's natural first instinct is to change the orientation of the segment, doing so requires implementing unnecessary changes in the code. The simplest approach is to rotate the domain. Figure-A II-3 shows examples of meshes with the segment at 10°, 30° and 60°. This method also facilitates the calculation and integration of forces over the segment.

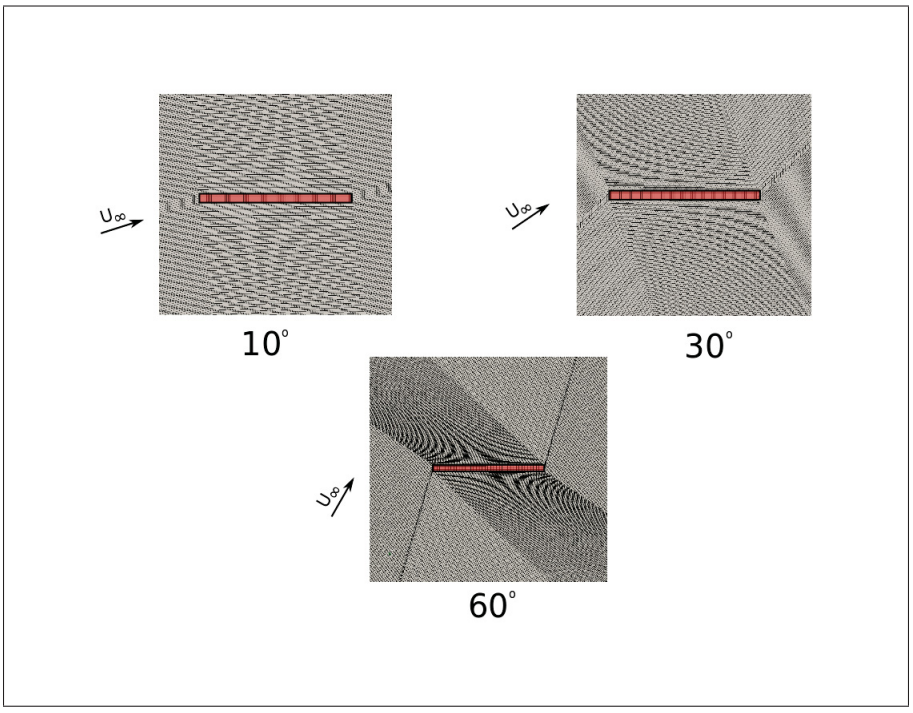

Figure-A II-3 Different domain/segment orientations.

1.2 Boundary conditions

Table-A II-1 shows the boundary conditions used in this case. In the 0 directory the files U and p are found. Notice that the empty patches must have the boundary condition set as empty. This will ensure that the simulation will be two-dimensional.

Table-A II-1 Boundary conditions

	Walls	**Inlet**	**Outlet**	**Empty**
Velocity	slip	fixedValue	zeroGradient	empty
Pressure	zeroGradient	zeroGradient	fixedValue	empty

1.3 Interpolation schemes

In regard to the interpolation schemes, the file fvSchemes is located in the system directory inside the case file. Special attention must be paid to the pressure and velocity schemes, which

must be set as presented in Figure-A II-4. As seen in the thesis, the correction terms are only conceived for a second order central differencing scheme. In OpenFOAM notation this is the linear scheme. If the pressure and velocity schemes are not properly set, the results will not be accurate.

```
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
//    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss upwind;
}

laplacianSchemes
{
    default         none;
    laplacian((1|A(U)),p) Gauss linear corrected;
}
```

Figure-A II-4 fvSchemes.

As seen in Figure-A II-4, for the velocity three different schemes can be used, namely a first order scheme **upwind**, and two second order schemes: **linearUpwind** (a mix between linear and upwind) and **SFCDV** (self-filtered central differencing for vector fields). If the **linearUpwind** scheme were to be used, simply replace **upwind** by **linearUpwind grad(U)**. Uncomment the grad(U) interpolation scheme. In this case, only a slight improvement is shown by the second order convection schemes.

1.4 New Dictionary

A set of new variables is needed to run the application; a new dictionary must be created. This dictionary is called ASDict and is located inside the constant directory. The easiest way to create ASDict is to copy another dictionary and change the name. In this dictionary certain variables must be defined, as shown in Figure-A II-5.

In the case of the 2D segment, $ymin$ and $ymax$ determine the location of the AS cell in the direction of the pressure jump. $xmin$ and $xmax$ set the location of the leading and trailing edges of the segment, while $zmin$ and $zmax$ are needed to select the cells in the dummy third

```
// * * * * * * * * * * * * * * * *
xmin 300;
xmax 400;
ymin 350;
ymax 351;
zmin 0.01;
zmax 0.49;
yvar 1;
Gamma 100;
// ****************************
```

Figure-A II-5 ASDict.

dimension required by OpenFOAM. To determine the loading of the segment, the total circulation $Gamma$ must be prescribed. $yvar$ is the length of the cell of the discretized segment for node selection purposes.

1.5 Other dictionaries

As seen in the case directory, other dictionaries are required as input for the application. An in-depth explanation on how to configure these files can be found in the user guide. Nevertheless a brief explanation is presented:

- **RASProperties**: Located in the constant directory. Since the turbulent term of the Navier-Stokes equations is not included this dictionary has no effect, but is required by OpenFOAM. For good measure the RASModel option is set to laminar and the turbulence to off.
- **RASProperties**: Located in the constant directory. Once again, this dictionary has no effect in the solution but is required. For good measure the viscosity is set to a very low value.
- **controlDict**: Located in the system directory. The number of iterations must be set to ensure a good convergence.

- **fvSolution**: Located in the system directory. Leave the default solvers, a convergence criterion can be set if necessary, as well as the relaxation factors.
- **sampleDict**: Located in the system directory. Used to take sample values for postprocessing.

2. Application particularities

In simpleFoamVD a new reference velocity field $Uref$ is included. This is done in order to calculate the perturbation flow. This field must be initialized in the file createFields.H as shown in Figure-A II-6. For both the new reference field and the velocity field (in time 0),boundary conditions must be initialized. This must be done taking the orientation of the domain/segment into account. For example if a flow of 10 m/s is desired and the domain is oriented at 0°, then the velocity vector would be $(10, 0, 0)$. If the domain were oriented at 30° then the velocity vector would be $(8.66025403784, 5, 0)$.

```
volVectorField Uref
(
        IOobject
        (
                "Uref",
                runTime.timeName(),
                mesh,
                IOobject::NO_READ


        ),
        mesh,
        dimensionedVector("Uref",dimVelocity,Foam::vector(8.66025403784, 5, 0)
);
```

Figure-A II-6 createFields.H.

When this application was implemented, one major inconvenience was encountered. The OpenFOAM function **mesh.findCell()** does not work in parallel. This function is needed in o r d e r to obtain the average velocity. Therefore all simulations must be performed in one processor. To run the application, simply go to the case file and type simpleFoamVD in the terminal.

APPENDIX III

SIMPLEFOAMWT TUTORIAL

The following is a tutorial in how to use the application simpleFoamWT. This application works for a finite wing in uniform translation.

1. Solver Inputs

First a case directory must be created. Figure-A III-1 shows an example of the case directory for the wing in translation.

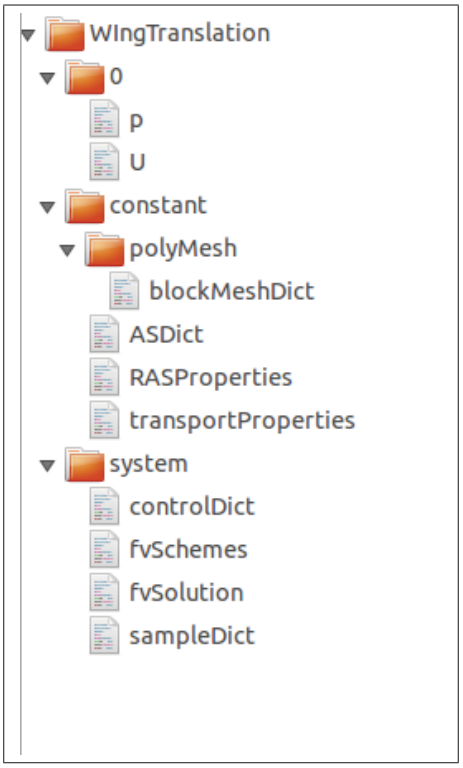

Figure-A III-1 Case directory.

This is a standard OpenFOAM case directory, with the necessary input files for the application to function correctly. Next, the most important input files will be reviewed.

1.1 Mesh handling

When creating the grid for this case, as usual four rows must be uniform in the normal direction. It is also important to refine the mesh within the vicinity of the AS.

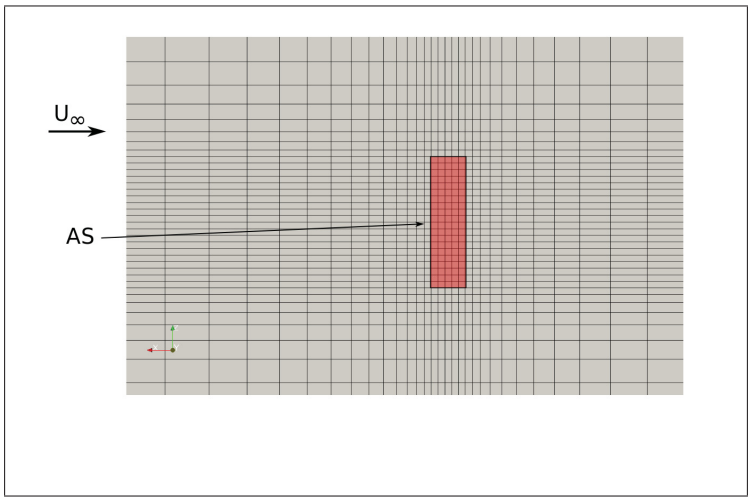

Figure-A III-2 Wing mesh refinement example.

Mesh refinement becomes an issue in the wing in translation case, because a decision has to be made regarding the resolution of the wing and the domain. This has to be done in order to obtain quality results. Mesh 2 of the application case was the most convenient to use. Nevertheless as long as the wing resolution of 13x32 is maintained, a coarser domain mesh does not significantly affect the results (as long as good refinement around the AS is kept).

1.2 Boundary conditions

In the 0 directory the files U and p are found. Table-A III-1 shows the boundary conditions for this case, which are shown in OpenFOAM notation.

1.3 Interpolation schemes

With regard to the interpolation schemes, the file fvSchemes is located in the system directory inside the case file. Special attention must be paid to the pressure and velocity schemes, which

Table-A III-1 Boundary conditions

	walls	**inlet**	**outlet**
Velocity	slip	fixedValue	zeroGradient
Pressure	zeroGradient	zeroGradient	fixedValue

must be set as presented in Figure-A III-3. As seen in the thesis, the correction terms are only conceived for a second order central differencing scheme. In OpenFOAM notation this is the linear scheme. If the pressure and velocity schemes are not properly set, the results will not be accurate.

```
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         Gauss linear;
    div(phi,U)      bounded Gauss linearUpwindV grad(U);

}

laplacianSchemes
{
    default         Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
}
```

Figure-A III-3 fvSchemes.

As seen in Figure-A III-3, for the velocity three different schemes can be used, namely a second order scheme **linearUpwind** (a mix between linear and upwind), another second order scheme **SFCDV** (self-filtered central differencing for vector fields) and a first other scheme **upwind**. In this case the second order schemes show substantial superiority in result quality, especially when the induced angle is compared with Prandtl's lifting line.

1.4 New Dictionary

A set of new variables is needed to run the application; a new dictionary must be created. This dictionary is called ASDict and is located inside the constant directory. The easiest way to

create ASDict is to copy another dictionary and change the name. In this dictionary certain variables must be defined, as shown in Figure-A III-4.

```
// * * * * * * * * * * * * * * * * * * *

/* LIMITS OF THE AS*/
xmin 100;

xmax 102.5;

ymin 150;

ymax 150.5;

zmin 145;

zmax 155;

/* PITCH ANGLE*/

Beta 0.087266462;

/* Uinf*/

Vo 50;

/* CELL SIZES*/
xvar 0.5;

yvar 0.5;

// ************************************
```

Figure-A III-4 ASDict.

In the case of the wing in translation, $ymin$ and $ymax$ determine the location of the AS cell in the direction of the pressure jump. $xmim$ and $xmax$ set the location of the leading and trailing edges of the segment, while $zmin$ and $zmax$ are the spanwise limits of the wing. To determine the loading of the wing the incoming flow Vo is required, as well as the wing pitch angle $Beta$. $xvar$ and $yvar$ are the length of the cell in X and Y direction inside the discretized wing for node selection purposes.

1.5 Other dictionaries

As seen in the case directory, other dictionaries are required as input for the application. An in-depth explanation on how to configure these files can be found in the user guide, but a brief description is presented below:

- **RASProperties**: Located in the constant directory. Since the turbulent term of the Navier-Stokes equations is not included, this dictionary has no effect, but is required by OpenFOAM. For good measure the RASModel option is set to laminar, and the turbulence to off.
- **RASProperties**: Located in the constant directory. Once again, this dictionary has no effect in the solution, but is required. For good measure the viscosity is set to a very low value.
- **controlDict**: Located in the system directory. The number of iterations must be set to ensure a good convergence.
- **fvSolution**: Located in the system directory. Leave the default solvers; a convergence criterion can be set if necessary, as can the relaxation factors.
- **sampleDict**: Located in the system directory. Used to take sample values for post-processing.

2. Application particularities

In this application, the parabolic distribution was used to calculate the velocity discontinuity. This distribution was used only in cases with a taper ratio of 1. Only the second term of the Δv_p equation was implemented:

$$\Delta v_p = \underbrace{\frac{6\Gamma}{c^4}\frac{\delta c}{\delta y}(x_p^3 - cx_p^2)}_{\text{not implemented}} + \frac{1}{c^3}\frac{\delta \Gamma}{\delta y}(3cx_p^2 - 2x_p^3) \qquad \text{(A III-1)}$$

If a wing of another taper ratio is desired the first term has to be added in deltaU.H. For this a new variable that measures the chord gradient must be added to the application code.

Regarding the mesh, the only option is to use an odd number of cells to discretize the segments. This is because the angle α_i has to be evaluated in the middle of the chord. Also, for grids

where the segments are discretized with more than 15 nodes, it is recommended to map pre-existing results from another simulation with a coarser resolution. This must be done before proceeding, as the finer the resolution becomes, the more unstable the simulation becomes.

At this point this application only works by calculating the ideal lift coefficient. However, if desired experimental data from a specific airfoil(s) can be added to the application. To run the application type simpleFoamWT in the terminal from the case file.

APPENDIX IV

SIMPLEFOAMBR TUTORIAL

This tutorial is for the application simpleFoamBR. This application is implemented using the Actuator Surface model for 3D cases applied to wind turbines. This application uses a rotating frame of reference.

1. Solver Inputs

First a case directory must be created. Figure-A IV-1 shows an example of the case directory for the wind turbine.

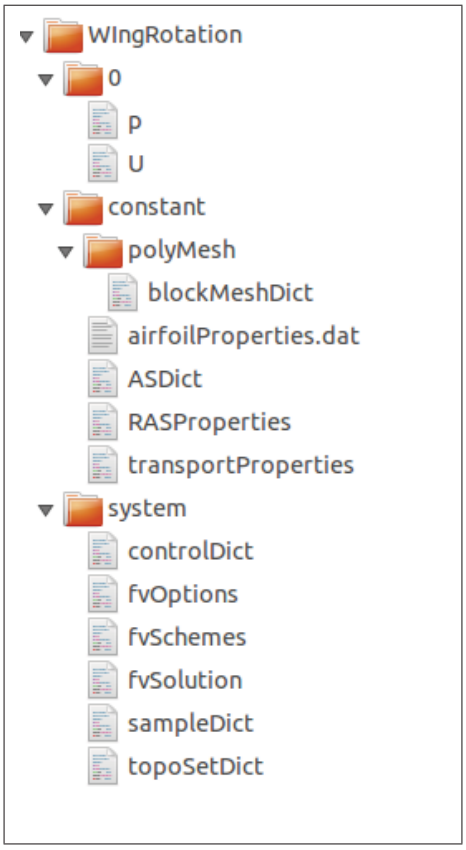

Figure-A IV-1 Case directory.

This is a standard OpenFOAM case directory, with the necessary input files for the application to function correctly. Next, the most important input files will be reviewed.

1.1 Mesh handling

In this case the mesh consists of a half cylinder, where the AS is facing the flow. A refinement must be made around the AS as shown in Figure-A IV-2. In order for both sides of the mesh to have the same amount of cells, the wing sections have 18 cells, but only 17 are selected for the AS. The middle cell is exactly on $\theta = \frac{\pi}{2}$ to evaluate the angle α_i.

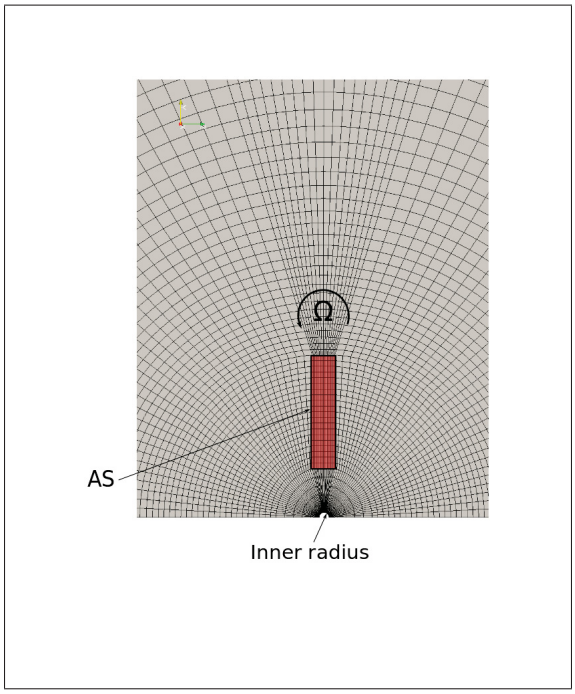

Figure-A IV-2 Close-up of blade & inner radius.

Another characteristic of the mesh is the inner radius. This radius is necessary because since OpenFOAM only works in Cartesian coordinates, a division by zero will occur at the centre of rotation if the inner radius is not kept. The mesh is used for a two-bladed turbine. Only half of the domain is modelled under the assumption that since both blades are identical, there is

a periodicity of the flow in both sides of the domain. In the case of a three-bladed turbine, a similar approach can be used as shown in Figure-A IV-3.

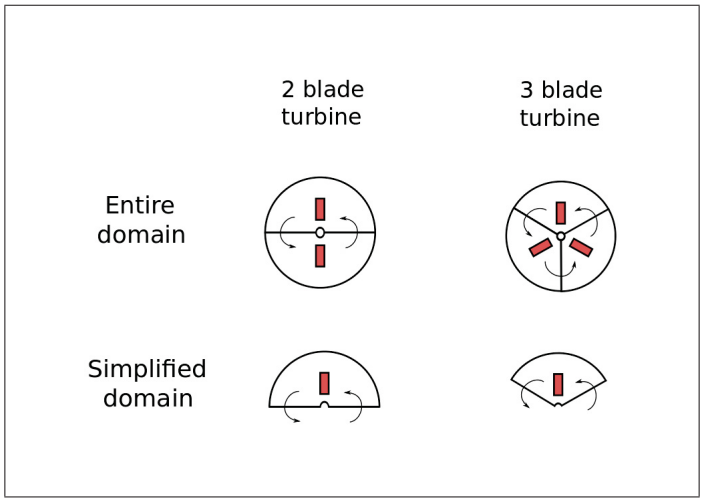

Figure-A IV-3 Different domain configurations.

In OpenFOAM, two patches can act as if they were connected using the cyclic boundary condition. The cyclic boundary condition connects a master patch (plane_half0) to a slave patch (plane_half1). Nevertheless, cyclic alone will assume that one patch is in front of the other. This does not work with a rotational flow. Therefore, when creating the patches, the option transform rotational must be used in the dictionary blockMeshDict. This is done by adding the lines shown in Figure-A IV-4.

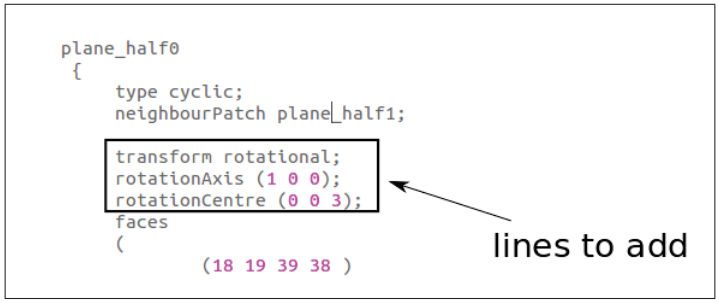

Figure-A IV-4 Lines to add for rotational option.

Where the axis and the centre of rotation must be specified. Figure-A IV-5 shows the effect of the transform rotational option over the cyclic patch.

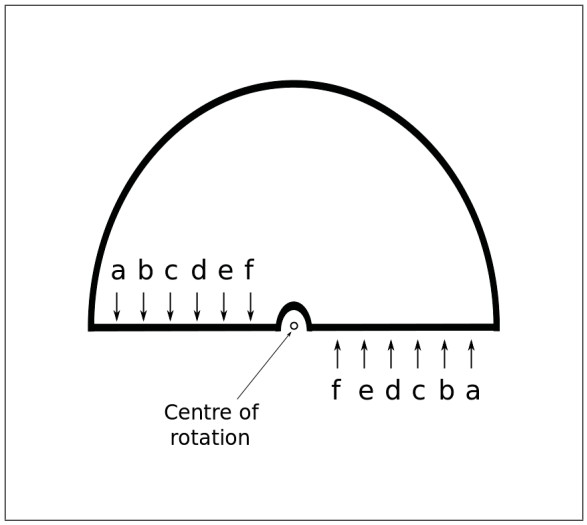

Figure-A IV-5 Cyclic patch with rotational option.

1.2 Boundary conditions

In the 0 directory the files U and p are found. Table-A IV-1 shows the boundary conditions for this case in OpenFOAM notation.

Table-A IV-1 Boundary conditions

	Velocity	**Pressure**
Walls	slip	zeroGradient
Inlet	fixedValue	zeroGradient
Outlet	zeroGradient	fixedValue
Plane_half0	cyclic	cyclic
Plane_half1	cyclic	cyclic
InnerRadius	slip	zeroGradient

1.3 Interpolation schemes

With regard to the interpolation schemes, the file fvSchemes is located in the system directory inside the case file. Special attention must be paid to the pressure and velocity schemes, which must be set as presented in Figure-A IV-6. As seen in the thesis, the correction terms are only conceived for a second order central differencing scheme. In OpenFOAM notation this is the linear scheme. If the pressure and velocity schemes are not properly set, the results will not be accurate.

```
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         Gauss linear;
    div(phi,U)      bounded Gauss linearUpwindV grad(U);

}

laplacianSchemes
{
    default         Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
}
```

Figure-A IV-6 fvSchemes.

The case of the wing in translation has provided valuable lessons. A medium mesh resolution is used. For the velocity field only the second order scheme **linearUpwind** (a mix between linear and upwind) is used, as it provides a better convergence than **SFCDV**.

1.4 New Dictionary

A set of new variables is needed to run the application; a new dictionary must be created. This dictionary is called ASDict and is located inside the constant directory. The easiest way to create ASDict is to copy another dictionary and change the name. In this dictionary certain variables must be defined, as shown in Figure-A IV-7.

In the case of the wind turbine, $xmin$ and $xmax$ determine the location of the AS cell in the direction of the pressure jump. $zmin$ and $zmax$ set the location of the leading and trailing

```
// * * * * * * * * * * * * * * * * * * *

/* LIMITS OF THE AS*/
xmin            1.264705882;

xmax             1.269411765;

ymin            0.18;

ymax            0.6;

zmin            2.96;

zmax            3.04;

/* OPERATION CHARACTERISTICS*/

Beta            2;

omega           73.3038285;

radius          0.6;

/* CELL SIZES*/
xvar            0.004705882353;

zvar            0.004705882353;

// ************************************
```

Figure-A IV-7 ASDict.

edges of the blade, while $ymin$ and $ymax$ are the spanwise limits of the blade. To determine the loading of the blade the rotational velocity $Omega$ is required, as well as the blade pitch angle $Beta$ and the $radius$. The variables $xvar$ and $zvar$ are the length of the cells in the X and Z direction inside the discretized blade, for node selection purposes.

1.5 Rotational frame

Since the method in use includes a rotational frame of reference, the following method was used to select the cells belonging to the rotational frame.

1.5.1 topoSetDict

First, the dictionary topoSetDict must be included in the system directory. topoSetDict must be configured as shown in Figure-A IV-8.

In this dictionary, p1 and p2 are the points defining the axis of rotation. A radius is required. When running this application all the cells inside the radius swept area along the axis of rotation

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

actions
(

    {
        name c0;
        type cellSet;
        action new;
        source cylinderToCell;
        sourceInfo
        {
        p1 (0.1 0 3); // start point on cylinder axis
        p2 (4 0 3); // end point on cylinder axis
        radius 2.67;
        }
    }

);

// ************************************************************
```

Figure-A IV-8 topoSetDict.

will be selected as part of the cell set. This is extremely important, as a layer of at least one cell must be left between the external boundary conditions and the rotating frame. This is shown in Figure-A IV-9.

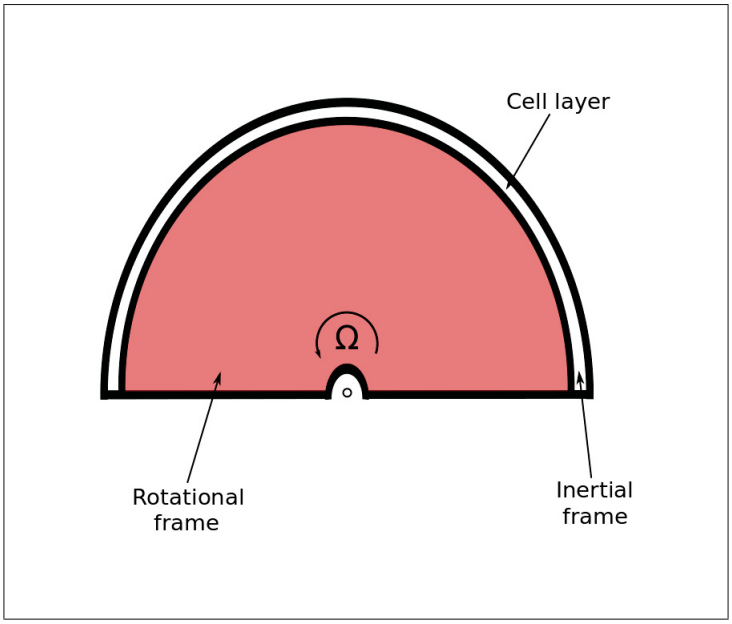

Figure-A IV-9 Illustration of inertial/rotational frame cell selection.

Once the topoSetDict is configured, run the application topoSet from the terminal in the case directory followed by setsToZones. This will transform the newly created set into a rotational zone.

1.5.2 fvOptions

Lastly, a fvOptions dictionary must be added to the system file; this dictionary must be configured as shown in Figure-A IV-10, where origin corresponds to the centre of rotation. The axis of rotation must be defined as well as the rotational velocity in rpm.

```
// * * * * * * * * * * * * * * * * * * * * * * * *

MRF1
{
    type            MRFSource;
    active          true;
    selectionMode   cellZone;
    cellZone        c0;

    MRFSourceCoeffs
    {
        origin      (0 0 3);
        axis        (1 0 0);
        omega       73.3038285;
    }
}

// ************************************************
```

Figure-A IV-10 fvOptions.

1.6 Experimental data

This application requires experimental data, specifically the lift coefficient. The file airfoilProperties.dat is located in the constant directory. Figure-A IV-11 shows an example for the profile NACA0012.

1.7 Other dictionaries

As seen in the case directory, other dictionaries are required as input for the application. An in-depth explanation on how to configure these files can be found in the user guide, but a brief description is presented below:

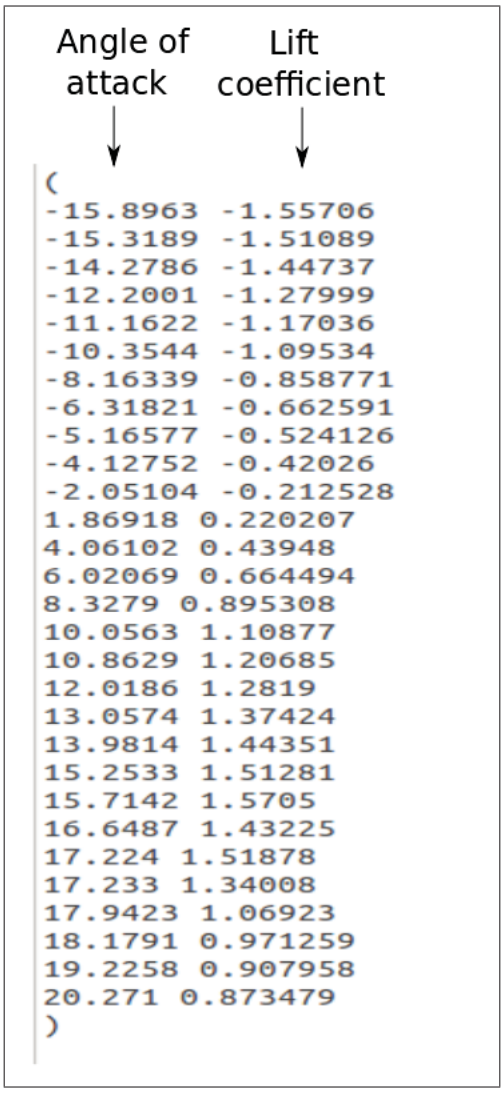

Figure-A IV-11 airfoilProperties.dat.

- **RASProperties**: Located in the constant directory. The turbulent term of the Navier-Stokes equations is required for the rotational frame to work, otherwise the application will crash immediately. This dictionary is configured by setting the RASModel option to laminar and the turbulence to off.
- **tranportProperties**: Located in the constant directory. An artificial viscosity is required to ensure the stability of the simulation. Different viscosity values should be used to determine which one gives better results. A value range between 1 and 10 times the air

viscosity was used. Results showed that 5 times the air viscosity was sufficient to ensure the stability of the solution.

- **controlDict**: Located in the system directory. The number of iterations must be set to ensure a good convergence.
- **fvSolution**: Located in the system directory. Leave the default solvers; a convergence criterion can be set if necessary, as can the relaxation factors.
- **sampleDict**: Located in the system directory. Used to sample values for post-processing.

2. Application particularities

In the case of the rotating frame, a layer of at least on cell must be left between the external boundaries and the frame. If the entire domain is selected as part of the rotating frame all the boundary conditions will be ignored when running the program. This application has been implemented to analyze a turbine with only one airfoil. If more airfoils are to be included, new .dat files must be added to the constant directory. New lists must be created in the createVariables.H file and assigned to the airfoil data during the force calculation routine. Lastly, to run the application type simpleFoamBR in the terminal from the case file.

LIST OF REFERENCES

Ainslie, John F. 1988. "Calculating the flowfield in the wake of wind turbines". *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, n° 1, p. 213–224.

Anderson John, D. 2005. "Fundamentals of aerodynamics".

Breton, Simon-Philippe, C Sibuet Watters, and Christian Masson. 2010. "Using the actuator surface method to model the three-bladed MEXICO wind turbine". In *48th AIAA Aerospace Sciences Meeting, AIAA-2010-461, Orlando, FL.*

Calaf, Marc, Charles Meneveau, and Johan Meyers. 2010. "Large eddy simulation study of fully developed wind-turbine array boundary layers". *Physics of Fluids (1994-present)*, vol. 22, n° 1, p. 015110.

Claessens, MC. 2006. "The design and testing of airfoils for application in small vertical axis wind turbines". *Master of Science Thesis*.

Crespo, A., J. Hernández, and S. Frandsen. 1999. "Survey of modelling methods for wind turbine wakes and wind farms". *Wind Energy*, vol. 2, n° 1, p. 1–24.

Dobrev, I, F Massouh, and M Rapin. 2007. "Actuator surface hybrid model". In *Journal of Physics: Conference Series*. p. 012019. IOP Publishing.

Duque, Earl PN, Michael D Burklund, and Wayne Johnson. 2003. "Navier-Stokes and comprehensive analysis performance predictions of the NREL phase VI experiment". *Journal of Solar Energy Engineering*, vol. 125, n° 4, p. 457–467.

Goldstein, Sydney. 1929. "On the vortex theory of screw propellers". *Proceedings of the Royal Society of London*, vol. 123, n° 792.

Haans, Wouter, Tonio Sant, Gijs van Kuik, and Gerard van Bussel. 2008. "HAWT near-wake aerodynamics, Part I: axial flow conditions". *Wind Energy*, vol. 11, n° 3, p. 245–264.

Hansen, Martin Otto Laver, Jens Nørkær Sørensen, S Voutsinas, Niels Sørensen, and H Aa Madsen. 2006. "State of the art in wind turbine aerodynamics and aeroelasticity". *Progress in aerospace sciences*, vol. 42, n° 4, p. 285–330.

Jasak, Hrvoje. 1996. "Error analysis and estimation for the finite volume method with applications to fluid flows". PhD thesis, Imperial College London (University of London).

Johansen, Jeppe, NN Sorensen, JA Michelsen, and S Schreck. 2002. "Detached-eddy simulation of flow around the NREL phase-VI blade". In *ASME 2002 Wind Energy Symposium*. p. 106–114. American Society of Mechanical Engineers.

Kärrholm, Fabian Peng, 2008. *Numerical modelling of diesel spray injection, turbulence interaction and combustion*. Chalmers University of Technology.

Koning, C. 1935. Influence of the propeller on other parts of the airplane structure. *Aerodynamic Theory*, p. 366. Springer.

Leclerc, Christophe and Christian Masson. 2004. “ Toward blade-tip vortex simulation with an actuator-lifting surface model ”. *AIAA Paper*, .

Manwell, J.F., J.G. McGowan, and A.L. Rogers, 2010. *Wind Energy Explained: Theory, Design and Application*. Wiley.

Masson, Christian, Arezki Smaïli, and Christophe Leclerc. 2001. “ Aerodynamic analysis of HAWTs operating in unsteady conditions ”. *Wind Energy*, vol. 4, n° 1, p. 1–22.

Meyers, Johan and Charles Meneveau. 2010. “ Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer ”. *AIAA Paper*, vol. 827, p. 2010.

Panton, Ronald L. 1996. “ Incompressible flow ”.

Porté-Agel, Fernando, Hao Lu, and Yu-Ting Wu. 2010. “ A large-eddy simulation framework for wind energy applications ”. In *Fifth International Symposium on Computational Wind Engineering*.

Prospathopoulos, JM, ES Politis, K Rados, and P Chaviaropoulos. 2009. “ Enhanced CFD modelling of wind turbine wakes ”. In *Euromech 508 Colloquium on Wind Turbine Wakes*.

Rethore, Pierre-Elouan Mikael and Niels N Sørensen. 2008. “ Actuator disc model using a modified Rhie-Chow/SIMPLE pressure correction algorithm. Comparison with analytical solutions ”. In *2008 European Wind Energy Conference and Exhibition*.

Sanderse, B. 2009. “ Aerodynamics of wind turbine wakes ”. *Energy Research Center of the Netherlands (ECN), ECN-E–09-016, Petten, The Netherlands, Tech. Rep*.

Sanderse, B., S.P. van der Pijl, and B. Koren. 2011. “ Review of computational fluid dynamics for wind turbine wake aerodynamics ”. *Wind Energy*, vol. 14, n° 7, p. 799–819.

Sant, Tonio. 2007. “ Improving BEM-based aerodynamic models in wind turbine design codes ”.

Schepers, JG and H Snel. 2007. “ Model experiments in controlled conditions ”. *ECN Report*.

Shen, Wen Zhong, Jens Nørkær Sørensen, and Jianhui Zhang. 2007. “ Actuator surface model for wind turbine flow computations ”. In *2007 European Wind Energy Conference and Exhibition*.

Sibuet Watters, Christophe. 2009. “ Étude numérique du sillage tourbillonnaire d’une éolienne ”. PhD thesis, École de Technolgie Supérierure.

Sibuet Watters, Christophe and Christian Masson. 2010. “ Modeling of lifting-device aerodynamics using the actuator surface concept ”. *International journal for numerical methods in fluids*, vol. 62, n° 11, p. 1264–1298.

Snel, H. 1998. “Review of the present status of rotor aerodynamics”. *Wind Energy*, vol. 1, n° s 1, p. 46–69.

Snel, H, JG Schepers, and B Montgomerie. 2007. “The MEXICO project (Model Experiments in Controlled Conditions): The database and first results of data processing and interpretation”. In *Journal of Physics: Conference Series*. p. 012014. IOP Publishing.

Sørensen, Niels N, JA Michelsen, and S Schreck. 2002. “Navier–Stokes predictions of the NREL phase VI rotor in the NASA Ames 80 ft× 120 ft wind tunnel”. *Wind Energy*, vol. 5, n° 2-3, p. 151–169.

Svenning, Erik. 2010. “Implementation of an actuator disk in OpenFOAM Developed for OpenFOAM-1.5-dev”.

Vermeer, LJ, Jens Nørkær Sørensen, and A Crespo. 2003. “Wind turbine wake aerodynamics”. *Progress in aerospace sciences*, vol. 39, n° 6, p. 467–510.

Watters, Christophe Sibuet and Christian Masson. 2007. “Recent advances in modeling of wind turbine wake vortical structure using a differential actuator disk theory”. In *Journal of Physics: Conference Series*. p. 012037. IOP Publishing.

Whale, J, KH Papadopoulos, CG Anderson, CG Helmis, and DJ Skyner. 1996. “A study of the near wake structure of a wind turbine comparing measurements from laboratory and full-scale experiments”. *Solar energy*, vol. 56, n° 6, p. 621–633.